Real Dataset Analysis

This analysis compares distinct approaches for modelling competing risks using a high-dimensional dataset on non-muscle-invasive bladder cancer from Dyrskjøt et al. (2004). This dataset includes gene expression (1,381 variables), clinical information, records of death (for bladder cancer, other causes, and not registered), and the time to the event (progression-free survival). In total, the dataset contains 404 observations.

We will compare the performance of this multinomial model against a penalized binary Cox proportional hazards model. In this common approach, a separate model is fitted for each event type, treating all other competing events as censored. This dataset has previously been analyzed by Tapak et al. (2015) using cause-specific Cox models with LASSO, elastic net, SCAD, and SICA penalizations to identify prognostic gene signatures.

1 Preprocess data

First, the two gene expression files were loaded, combined, and transposed to create a dataset where each row is a patient and each column is a gene probe. Next, the clinical data was loaded and cleaned by handling missing values and formatting key variables for survival analysis, including the survival `time` and a categorical `event` for competing outcomes. A minor adjustment was made, converting any survival times of 0 to 0.001 to prevent computational errors.

Finally, the gene expression and clinical datasets were merged by a unique `sample_id`. The resulting dataset was then filtere to include only the patient cohort used in the original study (Dyrskjøt et al. 2004) and to remove samples with missing data. As a result, the dataset kept 301 out of the 404 original observations.

```
# Process microarray data ---
bladder_fpd1 <- read_delim(here::here("notes_jmr/data/GSE5479_Final_processed_data_1.txt"
bladder_fpd2 <- read_delim(here::here("notes_jmr/data/GSE5479_Final_processed_data_2.txt"

bladder_fpd <- t(cbind(bladder_fpd1[, -1], bladder_fpd2[, -1])) %>%
  data.frame() %>%
  tibble() %>%
  bind_cols(tibble(sample_id = c(names(bladder_fpd1[, -1]),
                  names(bladder_fpd2[, -1])))) %>%
  set_names(c(bladder_fpd1$probe, "sample_id")) %>%
  clean_names() %>%
  select(sample_id, everything())

# Process clinical information
```

```
bladder_hd <- read_xls(here::here("notes_jmr/data/6517200/10780432ccr062940-sup-supplemen
  clean_names() %>%
  mutate(across(everything(), \(x)case_when(x == "-" ~ NA,
                                            T ~ x))) %>%
  transmute(
    sample_id = case_match(sample_id,
                           "1082-1" ~ "1082-1_DK",
                           "20421_S (91?)" ~ "20421_S",
                           .default = sample_id),
    country,
    # Survival time and event
    event = progression_0_no_progression_1_progression_to_t1_2_progression_to_t2,
    time = as.numeric(progression_free_survival),
    # time = as.numeric(follow_up_total),
    # Adjust time = 0
    time = ifelse(time == 0, 0.001, time),
    # Clinical variables
    age = as.numeric(age),
    female = if_else(str_trim(sex) == "F", 1, 0),
    progression = factor(progression_0_no_progression_1_progression_to_t1_2_progression_
    clinicalrisk = clinical_risk_1_high_risk_0_low_risk,
    followup = follow_up_months_from_tumor_to_last_visit_to_the_clinic_or_to_cystectomy,
    ## Reclassification of NA based on paper
    treatment = case_when(is.na(bcg_mmc_treatment) ~ "No treatment",
                          T ~ bcg_mmc_treatment),
    cystectomy = cystectomy,
    grade = reevaluated_who_grade_no_reevaluation,
    stage = reevaluated_pathological_disease_stage_no_reevaluation,
    # Identify samples used in the original paper's model training/validation
    progmodel = as.numeric(!is.na(samples_used_for_training_progression_classifier) | !is
 )


# Create complete bd
bladder_comp <- bladder_hd %>%
  mutate(sample_id = case_when(sample_id == "692-1" ~ paste0(sample_id,
                                                             "_",
                                                             country),
                               T ~ sample_id)) %>%
  select(-country) %>%
  full_join(bladder_fpd %>%
              #' Creating individual ID's for repeated ids.
              #' It assumes that the samples follow the same order as in
              #' supplementary file.
              mutate(sample_id = case_when(sample_id == "692-1...144" ~ "692-1_F",
                                           sample_id == "692-1...145" ~ "692-1_DK",
                                           T ~ sample_id)),
            by = join_by(sample_id)) %>%
  filter(progmodel == 1,
       !is.na(age),
       !is.na(female)) %>%
  select(-sample_id, -clinicalrisk, -cystectomy,
       -progmodel, -followup, -progression)
```

2 EDA

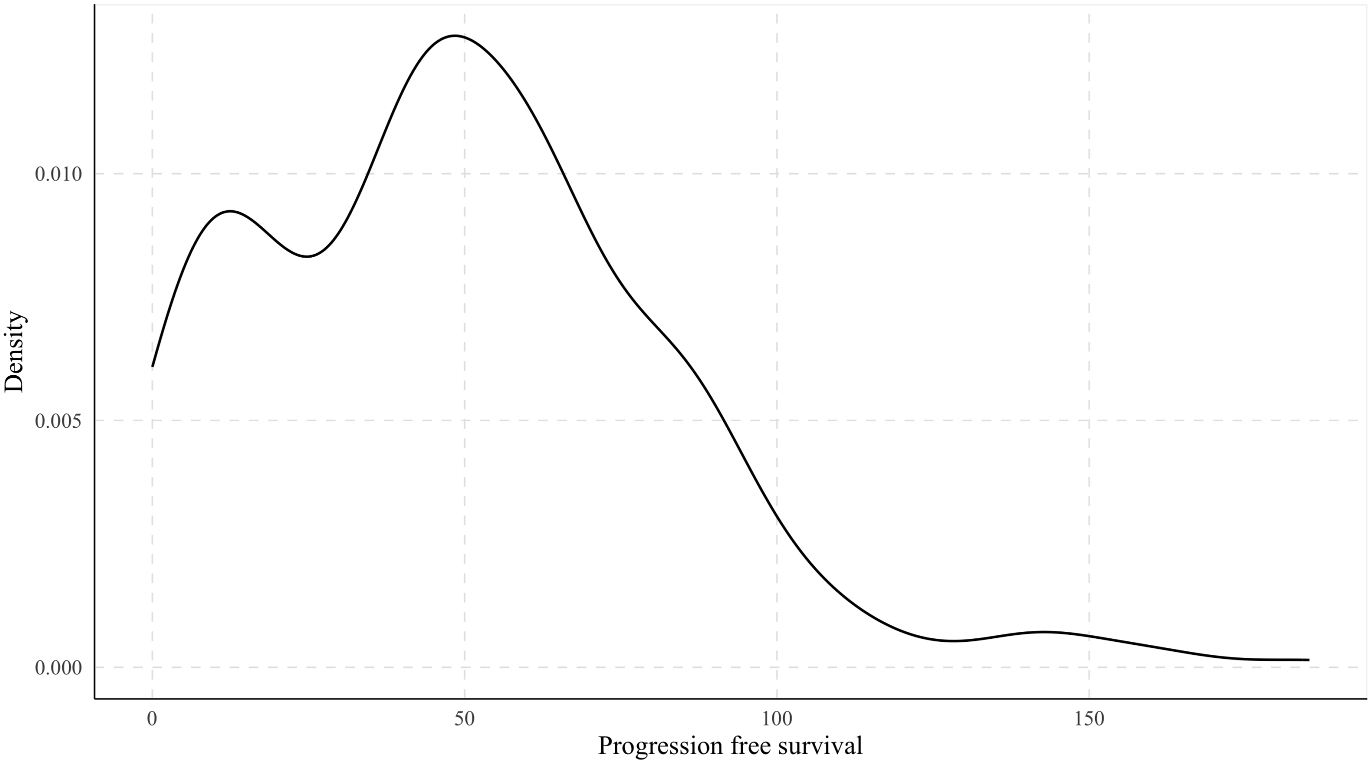
The majority of patients (227) were censored, 20 patients died from bladder cancer and 54 patients died from other causes.

```
#" Count events
bladder_comp %>%
  count(event) %>%
  adorn_totals()
```

```
#> event  n
#>    0 227
#>    1  20
#>    2  54
#> Total 301
```

The distribution of progression-free survival time is multi-modal, with a notable peak around 50 months and a long right tail. The average progression-free survival time across the cohort was 49.5 months. The population time plot visualizes the decrease in the at-risk population over the study's duration, with points indicating when progression events occurred.

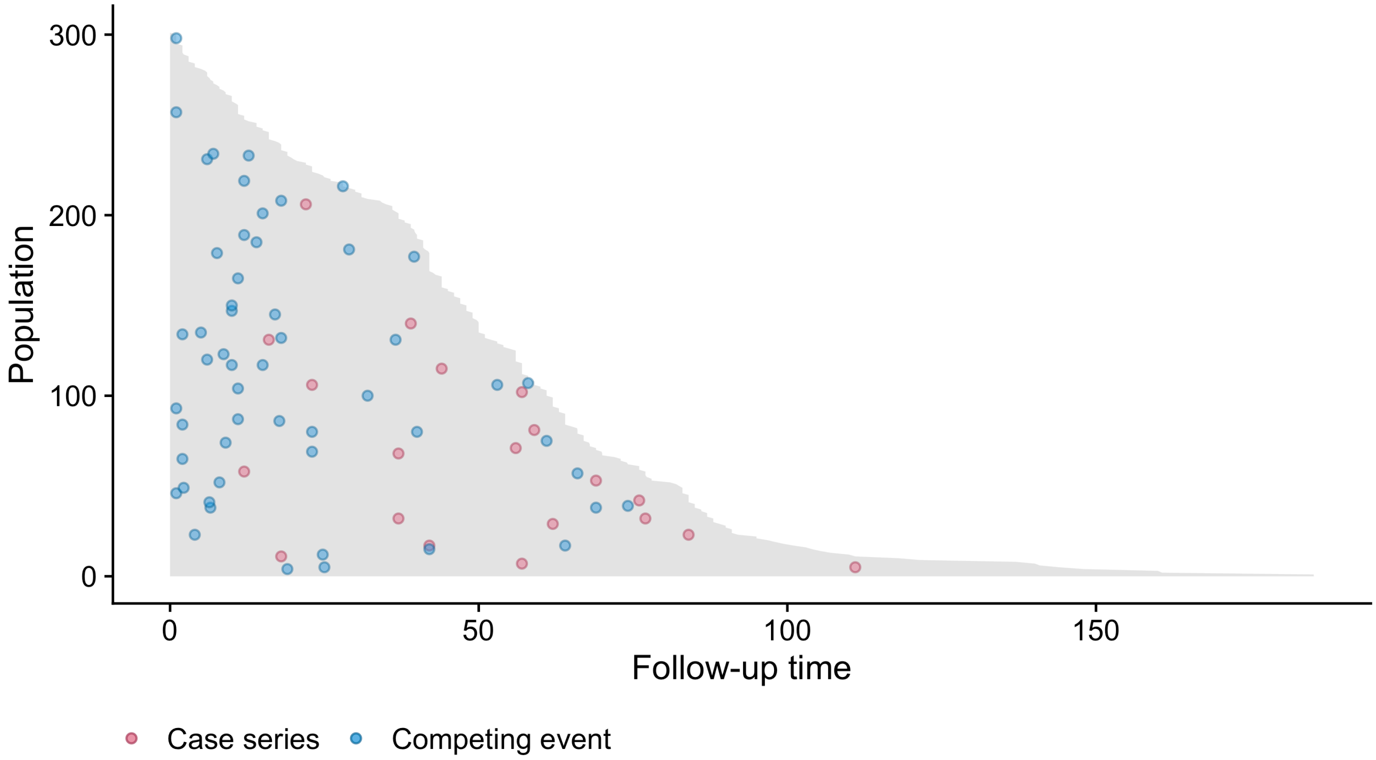
```
# Time
## Time dist
bladder_comp %>%
  ggplot(aes(x = time)) +
  geom_density() +
  labs(x = "Progression free survival",
      y = "Density")
```



```
## Mean time
mean(bladder_comp$event)
```

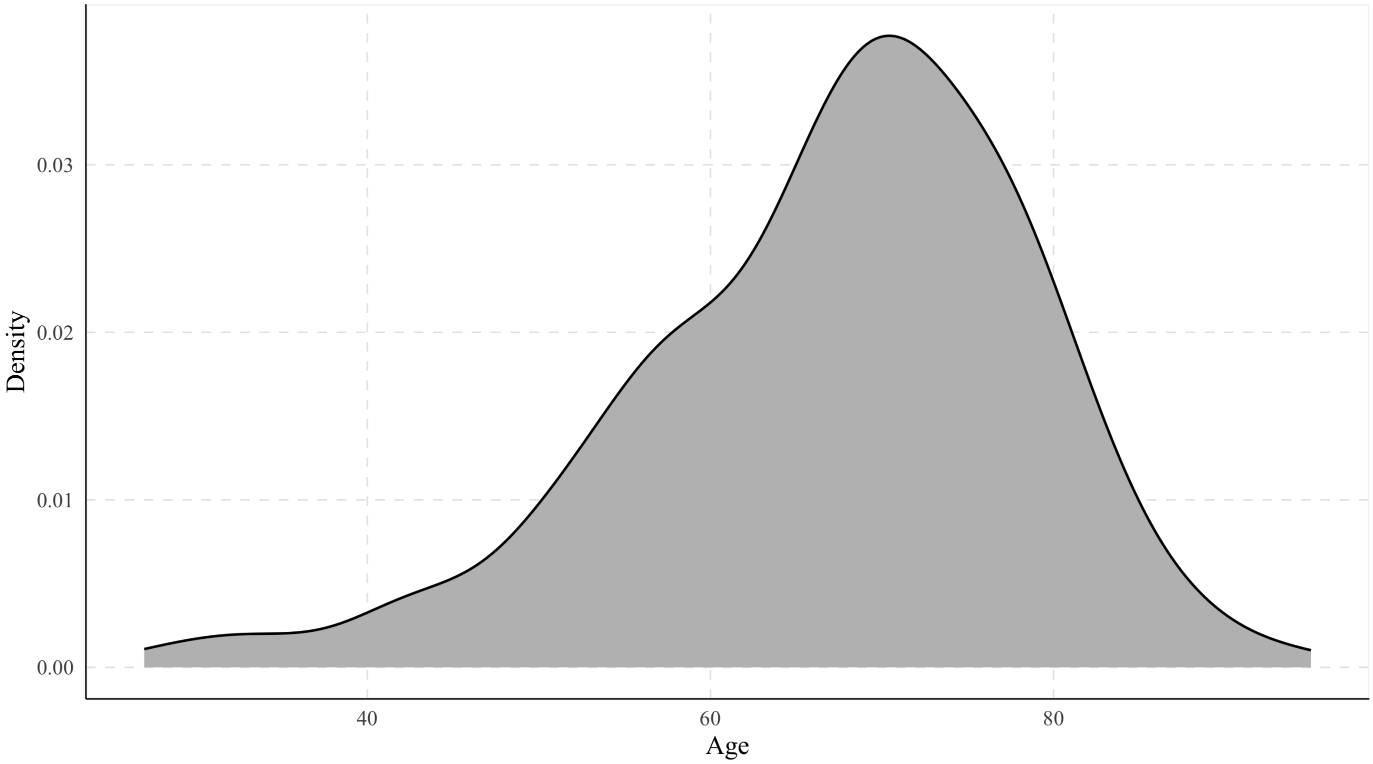
```
#> [1] 0.4252492
```

```
## Population time plot
plot(popTime(bladder_comp, "time", "event"),
    add.competing.event = TRUE,
   comprisk = TRUE)
```

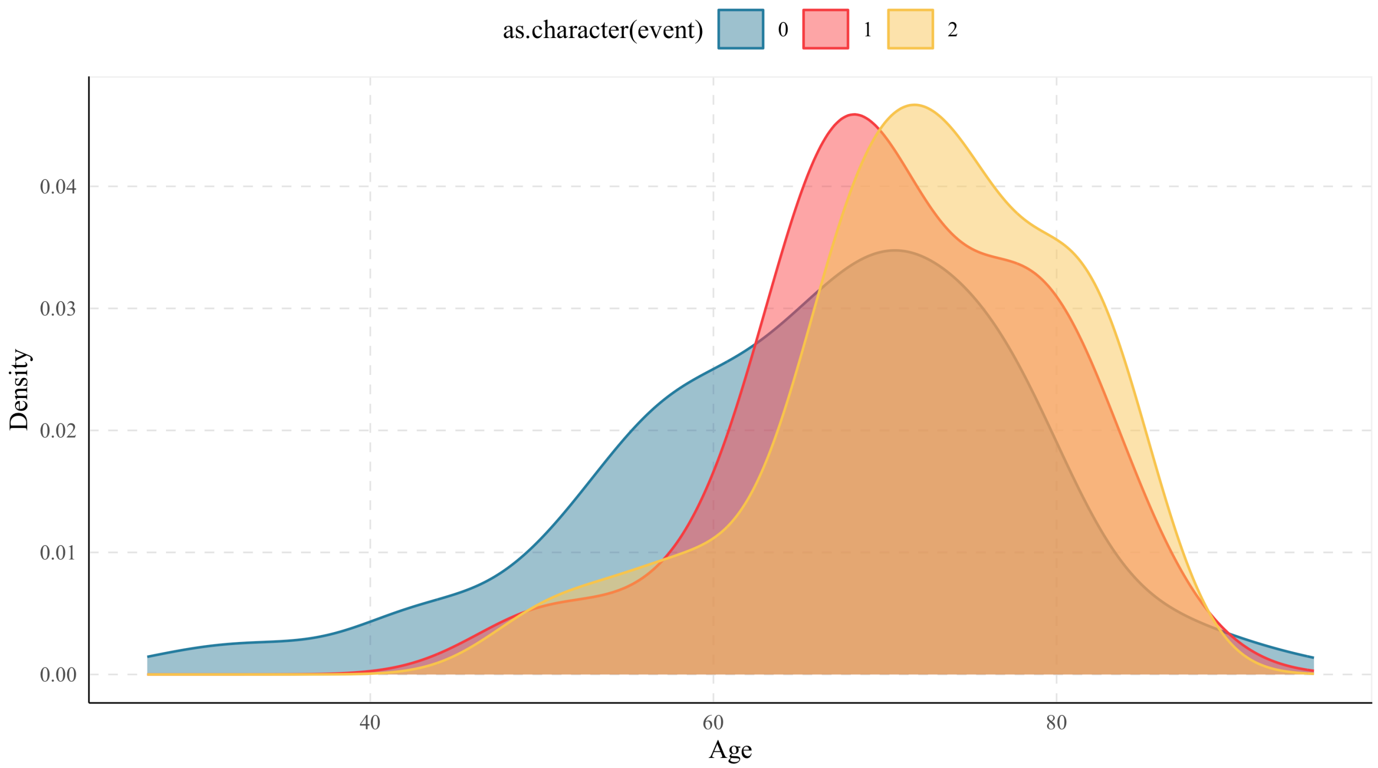


The patient population is predominantly male (241 males vs. 60 females), with an age distribution that peaks around 70 years.

```r
# Clinical variables
## Age dist
bladder_comp %>%
  ggplot(aes(x = age,)) +
  geom_density(fill = "grey") +
  labs(x = "Age",
     y = "Density")
```



```r
bladder_comp %>%
  ggplot(aes(x = age, colour = as.character(event),
           fill = as.character(event))) +
  geom_density(alpha = 0.5) +
  labs(x = "Age",
     y = "Density")
```



```
## Categorical var. levels
bladder_comp %>%
  select(female:stage) %>%
  mutate(across(everything(), as.character)) %>%
  pivot_longer(everything(),
               names_to = "Var",
               values_to = "Category") %>%
  group_split(Var, .keep = T) %>%
  map(~count(.,Var, Category) %>%
  adorn_totals())
```

```
#> [[1]]
#>    Var Category   n
#>  female        0 241
#>  female        1  60
#>   Total        - 301
#> 
#> [[2]]
#>   Var Category   n
#>  grade     HIGH 175
#>  grade    HIGH*   2
#>  grade      LOW  80
#>  grade  LOW OBS   1
#>  grade     LOW*  11
#>  grade   PUNLMP  32
#>  Total        - 301
#> 
#> [[3]]
#>   Var Category   n
#> stage     T1b   1
#> stage     pT1  23
#> stage    pT1*   2
#> stage    pT1a  54
#> stage    pT1b  47
#> stage     pTa 160
#> stage pTa obs   2
#> stage    pTa*  11
#> stage    pTis   1
#> Total       - 301
#> 
#> [[4]]
#>       Var     Category   n
#> treatment          BCG  73
#> treatment    BCG - MMC   4
#> treatment          MMC   5
#> treatment No treatment 219
#>   Total              - 301
```

Analysis of the clinical categories indicates that most patients had high-grade tumors and were at the pTa stage. A significant portion of the cohort (219 patients) did not receive intravesical BCG or MMC treatment.

Based on these categories, grade, stage, and treatment were recategorized as done in Ke, Bandyopadhyay, and Sarkar (2023). The PUNLMP (Papillary Urothelial Neoplasm of Low Malignant Potential) was classified as a low-grade. For the stage, all derived subcomments were removed, leaving only the pTa, pTis, and T1 categories. In comparison to Ke, Bandyopadhyay, and Sarkar (2023), they removed the observation with pTis stage. Lastly, treatment was classified as non versus either BCG or MMC. Another relevant difference from Ke, Bandyopadhyay, and Sarkar (2023) is that they categorized age; however, we do not have a clear reason to create arbitrary groups.

It is worth noting that the baseline categories were set to low for grade, pTa for stage, and none for treatment. This is especially relevant as the coefficients depend on the baseline group. Additionally, this can explain a different selection compared to previous studies using case-specific Cox models. For example, Tapak et al. (2015) used these models, but the preprocessing for the clinical variables and baseline categories was not found.

```r
# Adjust categories
bladder_comp_adj <- bladder_comp %>%
  mutate(grade = str_remove(grade, "\\*| OBS"),
      grade = case_when(str_detect(grade, "PUNLMP") ~ "LOW",
                        T ~ grade),
      grade = fct_relevel(grade, "LOW"),
      stage = case_when(str_detect(stage, "pTa") ~ "pTa",
                        str_detect(stage, "T1|pTis") ~ "T1",
                        T ~ stage),
      stage = fct_relevel(stage, "pTa"),
      # No treatment as base
      treatment = ifelse(str_detect(treatment, "BCG|MMC"), 1,
                   0))

bladder_comp_adj %>%
  select(female:stage) %>%
  mutate(across(everything(), as.character)) %>%
  pivot_longer(everything(),
          names_to = "Var",
          values_to = "Category") %>%
  group_split(Var, .keep = T) %>%
  map(~count(.,Var, Category) %>%
  adorn_totals())
```

```
#> [[1]]
#>     Var Category   n
#>  female        0 241
#>  female        1  60
#>   Total        - 301
#> 
#> [[2]]
#>    Var Category   n
#>  grade     HIGH 177
#>  grade      LOW 124
#>  Total        - 301
#> 
#> [[3]]
#>    Var Category   n
#>  stage       T1 128
#>  stage      pTa 173
#>  Total        - 301
#> 
#> [[4]]
#>        Var Category   n
#>  treatment        0 219
#>  treatment        1  82
#>      Total        - 301
```

```
## Categorical var. levels

bladder_comp_adj <- model.matrix(~ .,
                     data = bladder_comp_adj)[,-1] %>%
  as_tibble()
```

3 Model Fitting and Tuning

```
# Create a stratified 75/25 split
set.seed(1234)
split <- initial_split(bladder_comp_adj, prop = 0.75, strata = event)
```

```r
# Create training and testing data frames
train <- training(split)
test  <- testing(split)

# Verify the proportions
table(train$event) / nrow(train)
```

```
#> 
#>          0          1          2 
#> 0.75111111 0.07111111 0.17777778
```

```r
table(test$event) / nrow(test)
```

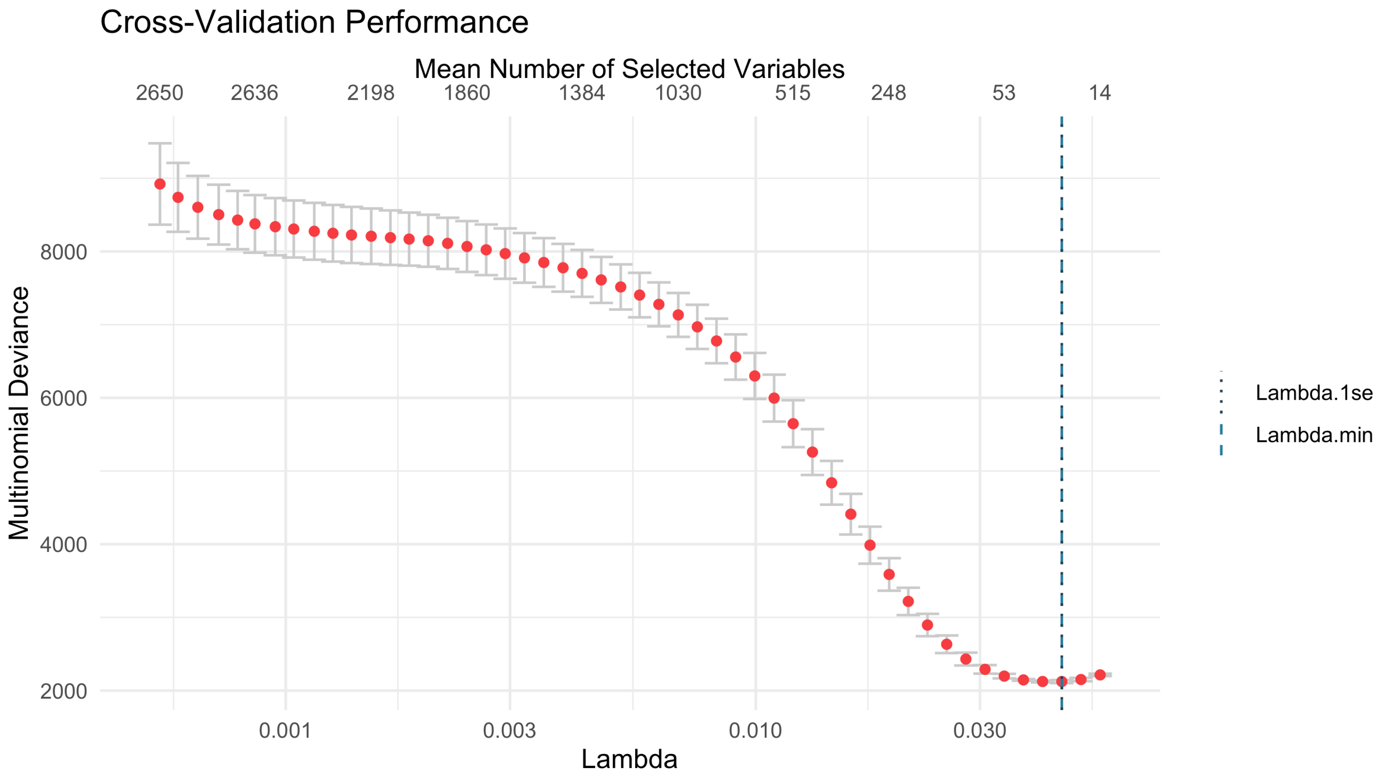
```
#> 
#>          0          1          2 
#> 0.76315789 0.05263158 0.18421053
```

3.1 Multinomial Elastic-Net Case-Base Model

```r
# Define the fitting function
mtool_fit_fun <- purrr::partial(cbSCRIP::MNlogistic,
                         niter_inner_mtplyr = 2,
                         maxit = 200,
                         tolerance = 1e-4,
                         learning_rate = 1e-4,
                         verbose = F,
                         save_history = F)

# Perform cross-validation on the training data

train %>%
  count(female, treatment, gradeHIGH, stageT1, event)
```

```
#> # A tibble: 33 x 6
#>    female treatment gradeHIGH stageT1 event     n
#>     <dbl>     <dbl>     <dbl>   <dbl> <dbl> <int>
#>  1      0         0         0       0     0    41
#>  2      0         0         0       0     1     5
#>  3      0         0         0       0     2     6
#>  4      0         0         0       1     0     2
#>  5      0         0         1       0     0    15
#>  6      0         0         1       0     1     1
#>  7      0         0         1       0     2     6
#>  8      0         0         1       1     0    40
#>  9      0         0         1       1     1     1
#> 10      0         0         1       1     2    15
#> # i 23 more rows
```

```
if(save){

  set.seed(1234)
  cv_multinom_enet <- cv_cbSCRIP(
    Surv(time, event) ~ .,
    cbind(train[,-(2:7), , drop = FALSE],
        train[,2:7, , drop = FALSE]),
    n_unpenalized = 7,
    alpha = 0.7,
    nfold = 5,
    nlambda = 50,
    fit_fun = mtool_fit_fun)

  plot(cv_multinom_enet)

  write_rds(cv_multinom_enet,
       here("paper",
                    "results",
                    glue("cv_multinom_enet.rds")))

  # set.seed(1234)
  #
  # fit.min <- cbSCRIP(
  #    Surv(time, event) ~ .,
  #    data = cbind(train[,-(2:7), , drop = FALSE],
  #        train[,2:7, , drop = FALSE]),
  #    # cb_data = cb_data_gen,
  #    n_unpenalized = 7,
  #    alpha = 0.5,
  #    lambda = .1,
  #    fit_fun = mtool_fit_fun,
  #    ratio = 50
  # )
  #
  # X_t <- model.matrix(~.,
  #          data = data.frame(cbind(fit.min$cb_data$covariates,
  #                          time = log(fit.min$cb_data$time))))
  #
  # X_t <- cbind(X_t[,-1], X_t[,1])
  #
  # p.fac <- rep(1, ncol(X_t))
  # p.fac[(1388+1-7):1388] <- 0
  #
  # fit.min$models_info[[1]]$convergence_pass
  # fit.min$models_info[[1]]$coefficients_sparse
  # loss <- map_vec(fit.min$models_info[[1]]$coefficients_history,
  #            ~calculate_penalized_multinomial_loss(
  #              .x,
  #               alpha = 0.7,
  #               lambda = 0.01,
  #               Y = fit.min$cb_data$event,
  #               X = X_t,
  #               offset = fit.min$cb_data$offset,
  #               penalty_weights = p.fac))
  #
  # if(min(loss) < min_loss) min_loss <- min(loss)
  #
  # tibble(iter = 1:length(loss),
  #      loss = loss) %>%
  #    mutate(loss_diff = loss - min_loss) %>%
  #    ggplot(aes(x = iter, y = loss_diff)) +
  #    geom_line() +
  #    scale_y_log10() +
  #    labs(y = "F(x)-F(x*)")

}


cv_multinom_enet <- readRDS(here("paper",
                     "results",
                     glue("cv_multinom_enet.rds")))


plot(cv_multinom_enet)
```



```
coefs_1 <- cv_multinom_enet$fit.min$coefficients[,1]

# selected variables for event 1.
select_coef_cbscrip_enet <- coefs_1[!same(coefs_1, 0)]
# select_coef_cbscrip_enet <- cv_multinom_enet$fit.min$coefficients[!same(cv_multinom_enet

sum(!same(cv_multinom_enet$fit.min$coefficients_sparse, 0))
```

```
#> [1] 14
```

```
## Number of non zero variables
length(select_coef_cbscrip_enet) - 7
```

```
#> [1] 0
```

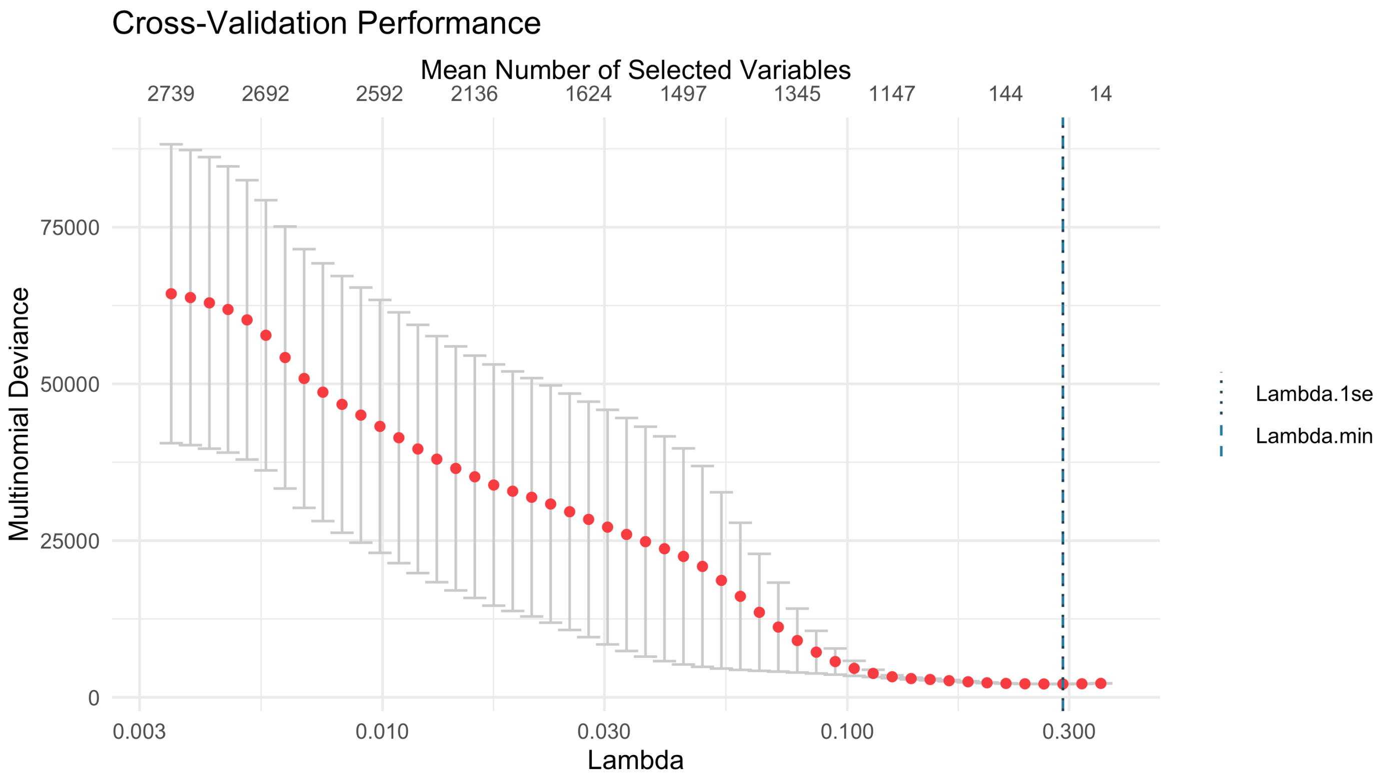
```
select_coef_cbscrip_enet
```

```
#>        age     female   treatment   gradeHIGH     stageT1   log(time)
#>  0.00717787 0.56207674  0.27328399 -0.09242324 -0.57209191 -1.59118091
#> (Intercept)
#> -2.06789493
```

3.2 Multinomial SCAD Case-Base Model

```
# Perform cross-validation on the training data
set.seed(1234)
if(save){
  cv_multinom_scad <- cv_cbSCRIP(
    Surv(time, event) ~ .,
    cbind(train[,-(2:7), , drop = FALSE],
        train[,2:7, , drop = FALSE]),
    n_unpenalized = 7,
    nfold = 5,
    nlambda = 50,
    lambda.min.ratio = 0.01,
    fit_fun = mtool_fit_fun,
    regularization = "SCAD",
    ratio = 50
  )

    saveRDS(cv_multinom_scad,
      here("paper",
        "results",
        glue("cv_multinom_scad.rds")))
}

cv_multinom_scad <- readRDS(here("paper",
        "results",
        glue("cv_multinom_scad.rds")))
```

```
plot(cv_multinom_scad)
```



```
# selected variables for event 1.
coefs_1_scad <- cv_multinom_scad$fit.min$coefficients[,1]

# selected variables for event 1.
select_coef_cbscrip_scad <- coefs_1_scad[!same(coefs_1_scad, 0)]
# select_coef_cbscrip_scad <- cv_multinom_scad$fit.min$coefficients[!same(cv_multinom_scac

select_coef_cbscrip_scad
```

```
#>         age      female   treatment   gradeHIGH     stageT1   log(time)
#>  0.00717538  0.56207347  0.27319473 -0.09234878 -0.57201788 -1.59116874
#> (Intercept)
#> -2.06778897
```

```
## Number of non zero variables
sum(!same(select_coef_cbscrip_scad, 0)) - 7
```

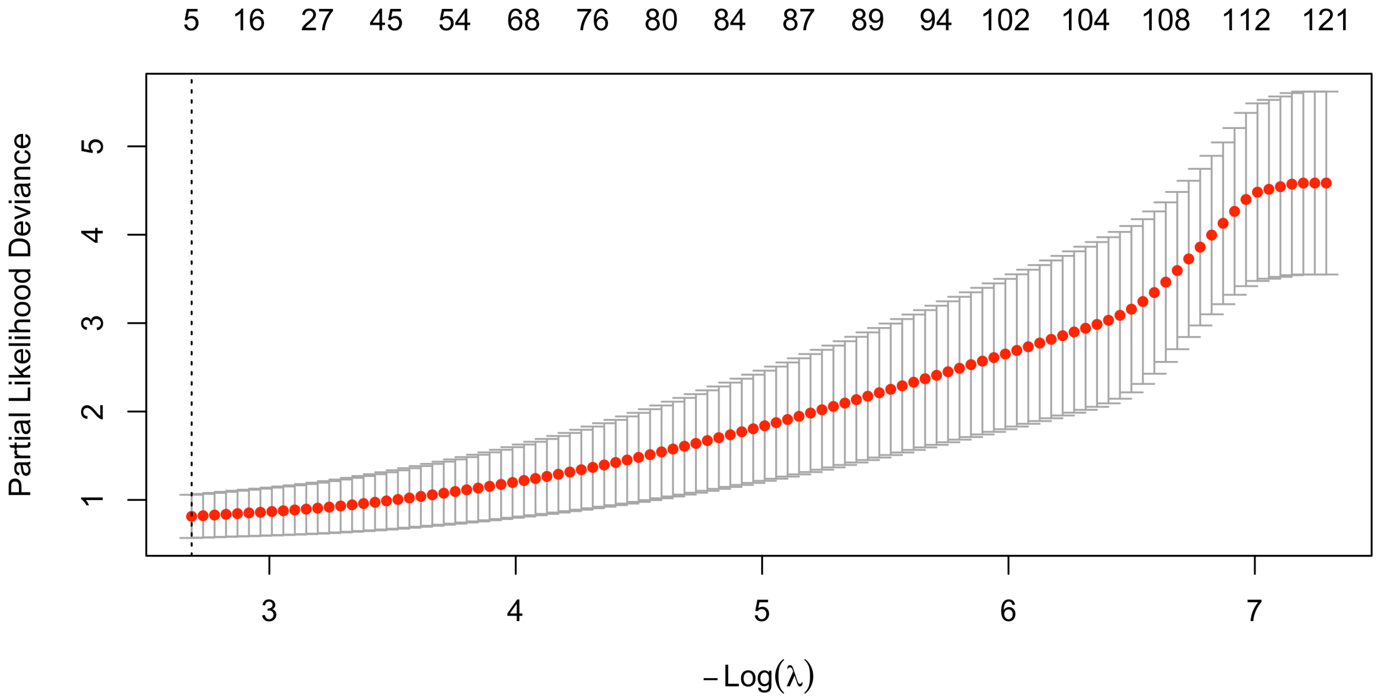
```
#> [1] 0
```

3.3 Cause-specific Cox models with Elastic-net

3.3.1 W/ Partial Likelihood Deviance

```
y <- Surv(time = train$time,
          event = as.numeric(train$event == 1))

x <- model.matrix(event ~ . -time,
```

```r
            data = train)

p.fac <- rep(1, ncol(x))
p.fac[1:6] <- 0

set.seed(1234)
cox_enet_mod <- cv.glmnet(x = x, y = y, family = "cox",
                 # family = "binomial",
                 penalty.factor = p.fac,
                 nfolds = 5,
                 alpha = 0.7,
                 thresh = 1e-9,
                 maxit = 1e9)

plot(cox_enet_mod)
```



```r
cc_enet_min <- coef(cox_enet_mod, s = cox_enet_mod$lambda.min)

select_vars_enet <- cc_enet_min@Dimnames[[1]][-1][cc_enet_min@i]

selected_coefs_enet <- cc_enet_min@x

names(selected_coefs_enet) <- select_vars_enet

selected_coefs_enet
```

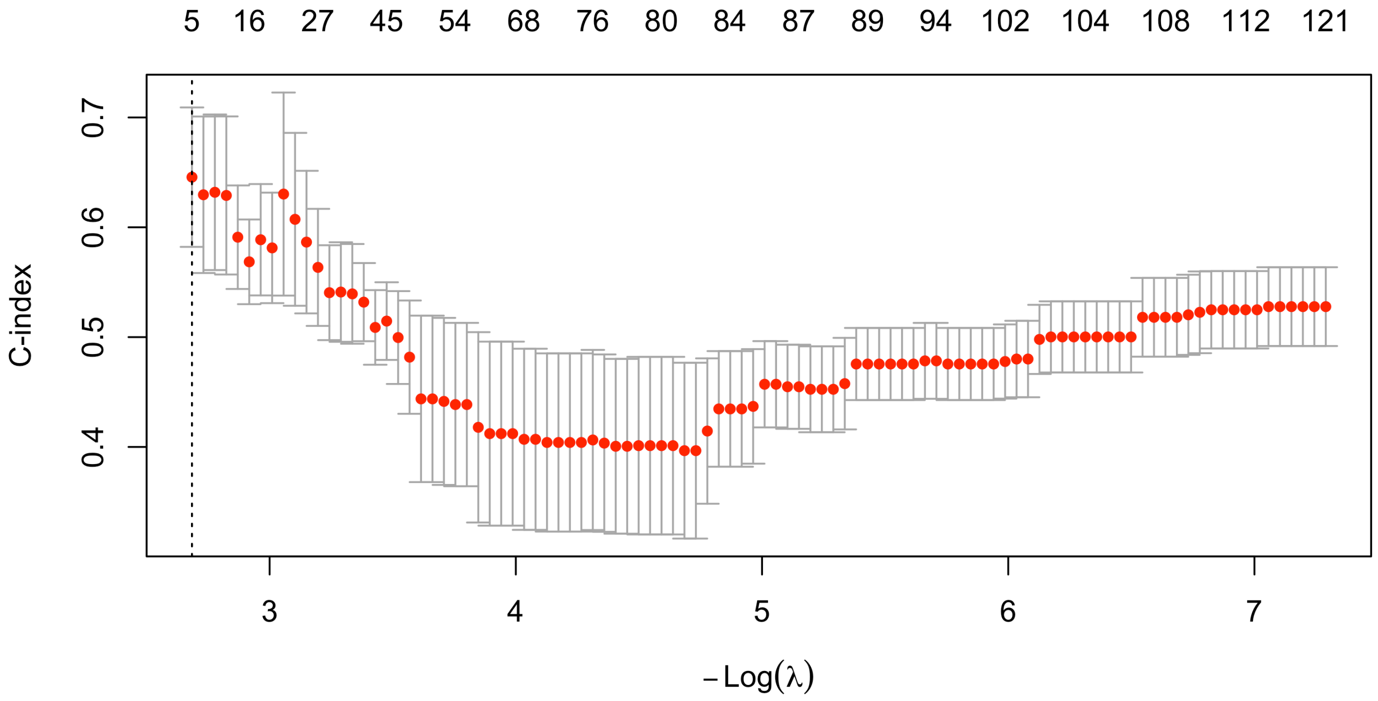
```
#>        age     female   treatment  gradeHIGH     stageT1
#> 0.04490115 0.96032176 -0.18137716 0.63784490 -1.56459726
```

```r
length(selected_coefs_enet)
```

```
#> [1] 5
```

3.3.2 W/ C Index

```
y <- Surv(time = train$time,
      event = as.numeric(train$event == 1))

x <- model.matrix(event ~ . -time,
            data = train)

p.fac <- rep(1, ncol(x))
p.fac[1:6] <- 0

set.seed(1234)
cox_enet_c_mod <- cv.glmnet(x = x, y = y, family = "cox",
                    # family = "binomial",
                    penalty.factor = p.fac,
                    nfolds = 5,
                    alpha = 0.7,
                    thresh = 1e-9,
                    type.measure = "C",
                    maxit = 1e9)

plot(cox_enet_c_mod)
```



```
cc_enet_c_min <- coef(cox_enet_c_mod, s = cox_enet_c_mod$lambda.min)

select_vars_enet_c <- cc_enet_c_min@Dimnames[[1]][-1][cc_enet_c_min@i]
```

```
selected_coefs_enet_c <- cc_enet_c_min@x

names(selected_coefs_enet_c) <- select_vars_enet_c

selected_coefs_enet_c
```

```
#>        age     female  treatment   gradeHIGH    stageT1
#>  0.04490115  0.96032176 -0.18137716  0.63784490 -1.56459726
```

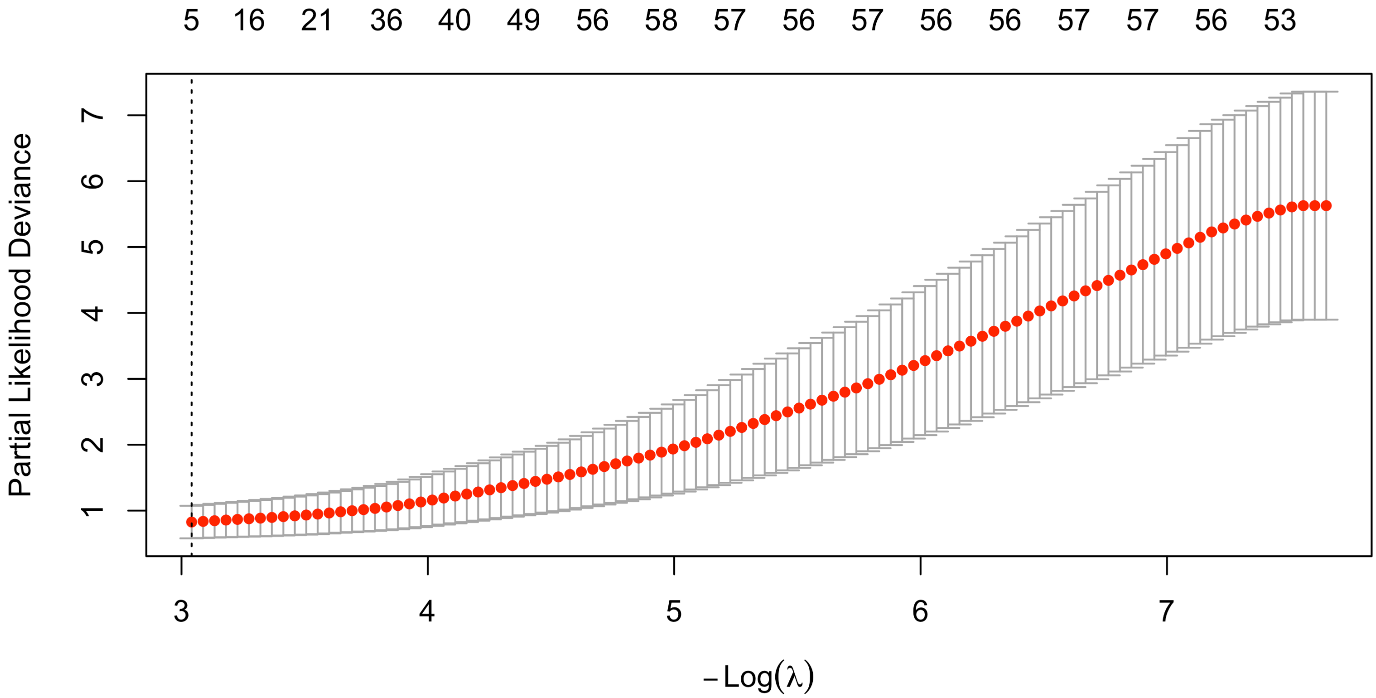
```
length(selected_coefs_enet_c)
```

```
#> [1] 5
```

3.4 Cause-specific Cox models with LASSO

3.4.1 W/ Partial Likelihood Deviance

```
y <- Surv(time = train$time,
      event = as.numeric(train$event == 1))

x <- model.matrix(event ~ . -time,
           data = train)

p.fac <- rep(1, ncol(x))
p.fac[1:6] <- 0

set.seed(1234)
cox_lasso_mod <- cv.glmnet(x = x, y = y, family = "cox",
                    # family = "binomial",
                    penalty.factor = p.fac,
                    nfolds = 5,
                    alpha = 1,
                    thresh = 1e-9,
                    maxit = 1e9)

plot(cox_lasso_mod)
```



```
cc_lasso_min <- coef(cox_lasso_mod, s = cox_lasso_mod$lambda.min)

select_vars_lasso <- cc_lasso_min@Dimnames[[1]][-1][cc_lasso_min@i]

selected_coefs_lasso <- cc_lasso_min@x

names(selected_coefs_lasso) <- select_vars_lasso

selected_coefs_lasso
```

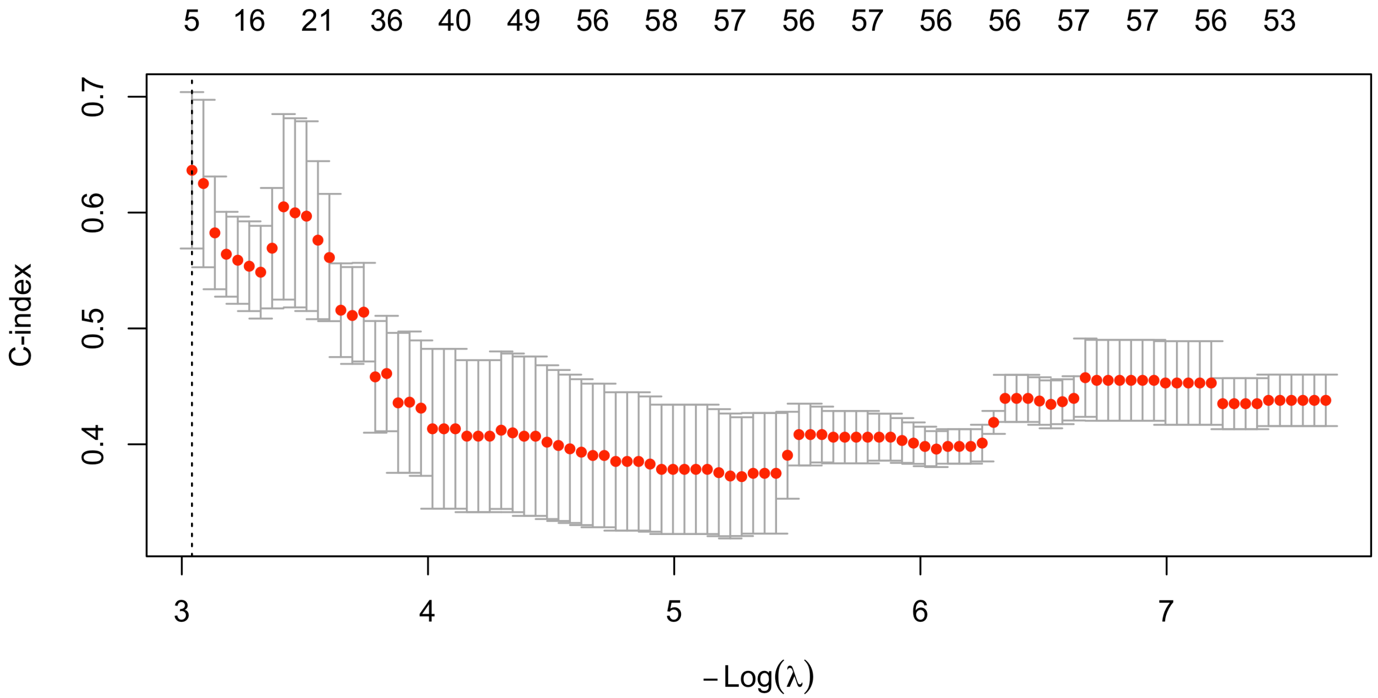
```
#>        age     female   treatment  gradeHIGH     stageT1
#> 0.04490115 0.96032176 -0.18137716 0.63784490 -1.56459726
```

```
length(selected_coefs_lasso)
```

```
#> [1] 5
```

3.4.2 W/ C Index

```
y <- Surv(time = train$time,
       event = as.numeric(train$event == 1))

x <- model.matrix(event ~ . -time,
            data = train)

p.fac <- rep(1, ncol(x))
p.fac[1:6] <- 0

set.seed(1234)
cox_lasso_c_mod <- cv.glmnet(x = x, y = y, family = "cox",
                  # family = "binomial",
                  penalty.factor = p.fac,
                  nfolds = 5,
                  alpha = 1,
                  thresh = 1e-9,
                  type.measure = "C",
                  maxit = 1e9)

plot(cox_lasso_c_mod)
```



```
cc_lasso_c_min <- coef(cox_lasso_c_mod, s = cox_lasso_c_mod$lambda.min)

select_vars_lasso_c <- cc_lasso_c_min@Dimnames[[1]][-1][cc_lasso_c_min@i]

selected_coefs_lasso_c <- cc_lasso_c_min@x

names(selected_coefs_lasso_c) <- select_vars_lasso_c

selected_coefs_lasso_c
```

```
#>        age     female   treatment   gradeHIGH    stageT1
#> 0.04490115  0.96032176 -0.18137716  0.63784490 -1.56459726
```

```
length(selected_coefs_lasso_c)
```

```
#> [1] 5
```

3.5 Summary

```
# No penalizar variables clínicas.
```

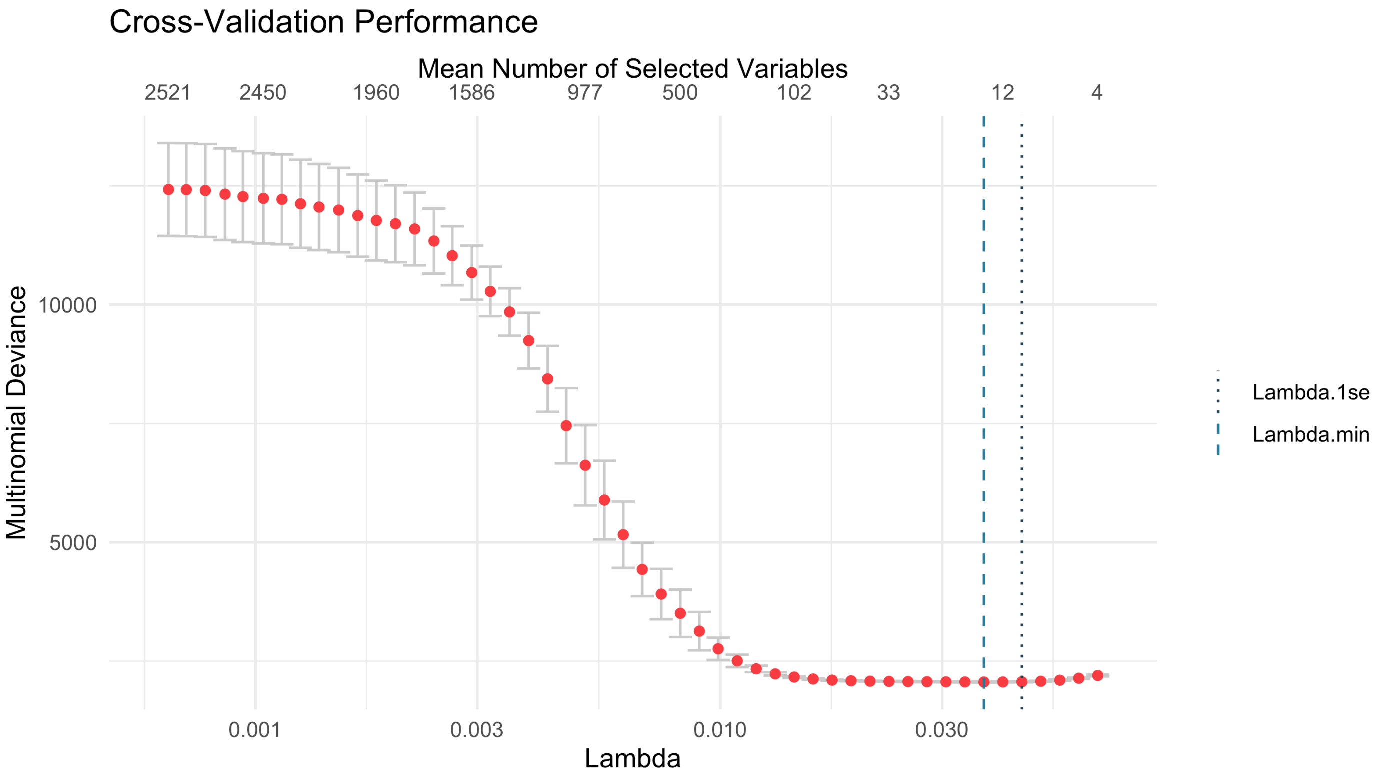
```
cbscrip_enet_coefs <- select_coef_cbscrip_enet[!(names(select_coef_cbscrip_enet) %in%
                            c("(Intercept)",
                             "log(time)"))]
cbscrip_scad_coefs <- select_coef_cbscrip_scad[!(names(select_coef_cbscrip_scad) %in%
                            c("(Intercept)",
                             "log(time)"))]

tibble(vars = names(selected_coefs_lasso),
     lasso = selected_coefs_lasso) %>%
  full_join(tibble(vars = names(selected_coefs_lasso_c),
             lasso_c = selected_coefs_lasso_c),
         by = join_by(vars))  %>%
  full_join(tibble(vars = names(selected_coefs_enet),
             enet = selected_coefs_enet),
         by = join_by(vars))  %>%
  full_join(tibble(vars = names(selected_coefs_enet_c),
             enet_c = selected_coefs_enet_c),
         by = join_by(vars))  %>%
  full_join(tibble(vars = names(cbscrip_enet_coefs),
             cbscrip_enet = cbscrip_enet_coefs),
         by = join_by(vars))  %>%
  full_join(tibble(vars = names(cbscrip_scad_coefs),
             cbscrip_scad = cbscrip_scad_coefs),
         by = join_by(vars)) %>%
  kbl()
```

vars	lasso	lasso_c	enet	enet_c	cbscrip_enet	cbscrip_scad
age	0.0449011	0.0449011	0.0449011	0.0449011	0.0071779	0.0071754
female	0.9603218	0.9603218	0.9603218	0.9603218	0.5620767	0.5620735
treatment	-0.1813772	-0.1813772	-0.1813772	-0.1813772	0.2732840	0.2731947
gradeHIGH	0.6378449	0.6378449	0.6378449	0.6378449	-0.0924232	-0.0923488
stageT1	-1.5645973	-1.5645973	-1.5645973	-1.5645973	-0.5720919	-0.5720179

4 Analysis whitout Clinical Variables

4.1 Multinomial Elastic-net Case-Base Model

```
# Perform cross-validation on the training data

if(save){

  set.seed(1234)
  mtool_fit_fun <- purrr::partial(cbSCRIP::MNlogistic,
                    niter_inner_mtplyr = 2,
                    maxit = 200,
                    tolerance = 1e-4,
                    learning_rate = 1e-4,
                    verbose = F,
                    save_history = F)
  cv_multinom_enet_nc <- cv_cbSCRIP(
    Surv(time, event) ~ .,
    train[,-(3:7), , drop = FALSE],
    alpha = 0.7,
    nfold = 5,
    nlambda = 50,
    fit_fun = mtool_fit_fun,
    ratio = 50)

  plot(cv_multinom_enet_nc)

  saveRDS(cv_multinom_enet_nc, here("paper",
          "results",
          glue("cv_multinom_enet_nc.rds")))

}


cv_multinom_enet_nc <- readRDS(here("paper",
                     "results",
                     glue("cv_multinom_enet_nc.rds")))


plot(cv_multinom_enet_nc)
```



```r
coefs_1_nc <- cv_multinom_enet_nc$fit.min$coefficients[,1]

# selected variables for event 1.
select_coef_cbscrip_enet_nc <- coefs_1_nc[!same(coefs_1_nc, 0)]

sum(!same(cv_multinom_enet_nc$fit.min$coefficients_sparse, 0))
```

```
#> [1] 12
```

```r
## Number of non zero variables
length(select_coef_cbscrip_enet_nc) - 2
```

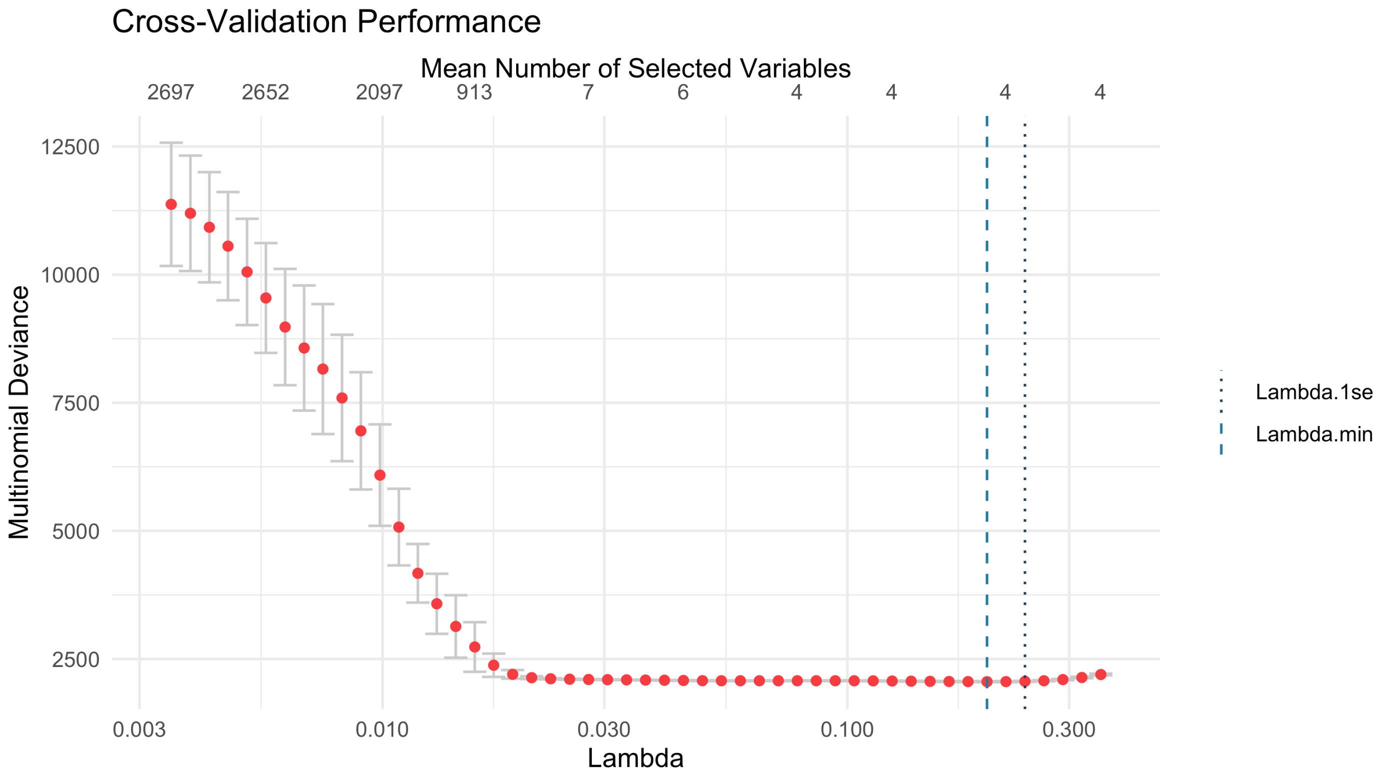
```
#> [1] 0
```

```r
select_coef_cbscrip_enet_nc
```

```
#>  log(time) (Intercept)
#>  -1.555563  -1.702721
```

4.2 Multinomial SCAD Case-Base Model

```r
# Perform cross-validation on the training data
if(save){

  set.seed(1234)
  cv_multinom_scad_nc <- cv_cbSCRIP(
    Surv(time, event) ~ .,
    cbind(train[,-(2:7), , drop = FALSE],
          train[,2:7, , drop = FALSE]),
    nfold = 5,
    nlambda = 50,
    fit_fun = mtool_fit_fun,
    regularization = "SCAD",
    ratio = 50
  )

   saveRDS(cv_multinom_scad_nc,
      here("paper",
        "results",
        glue("cv_multinom_scad_nc.rds")))
}

cv_multinom_scad_nc <- readRDS(here("paper",
        "results",
        glue("cv_multinom_scad_nc.rds")))

plot(cv_multinom_scad_nc)
```



```
# selected variables for event 1.
coefs_1_scad_nc <- cv_multinom_scad_nc$fit.min$coefficients[,1]

# selected variables for event 1.
select_coef_cbscrip_scad_nc <- coefs_1_scad_nc[!same(coefs_1_scad_nc, 0)]
# select_coef_cbscrip_scad <- cv_multinom_scad_nc$fit.min$coefficients[!same(cv_multinom_s

select_coef_cbscrip_scad_nc
```

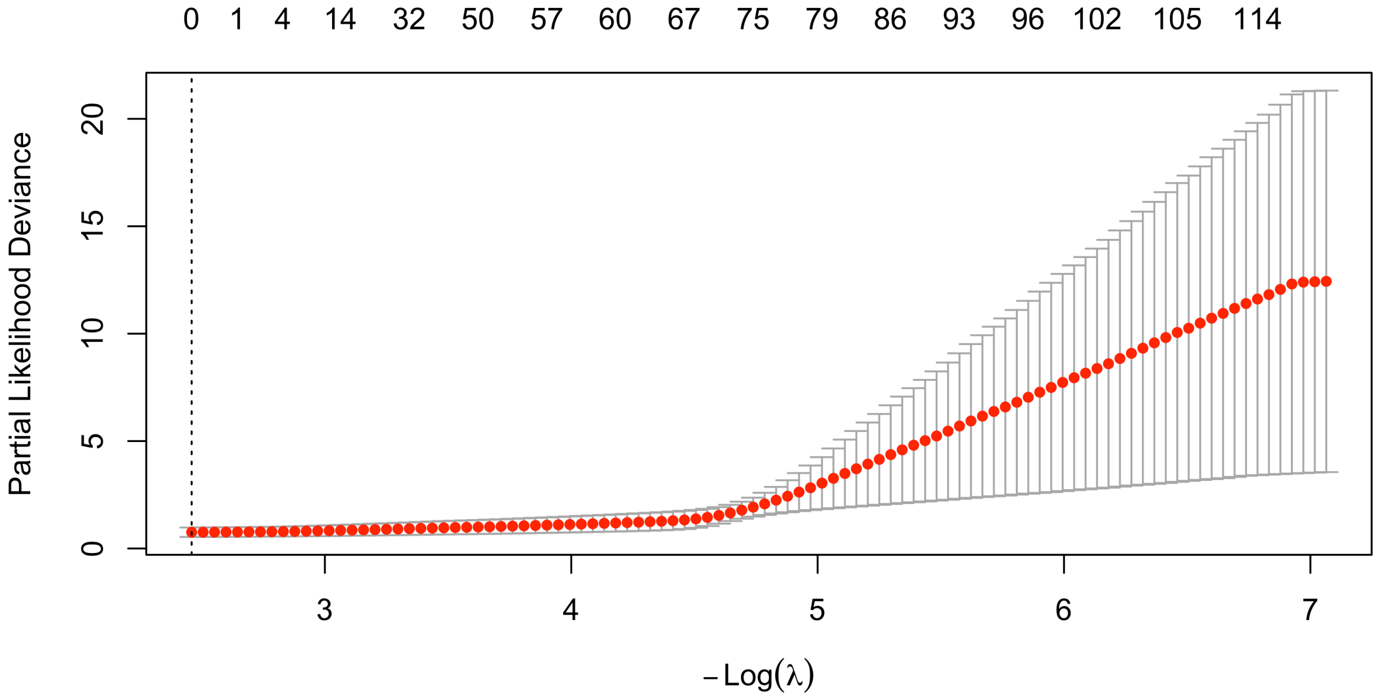
```
#>  log(time) (Intercept)
#>  -1.553673  -1.714267
```

```
## Number of non zero variables
sum(!same(select_coef_cbscrip_scad_nc, 0)) - 2
```

```
#> [1] 0
```

4.3 Cause-specific Cox models with Elastic-net

```
y <- Surv(time = train$time,
          event = as.numeric(train$event == 1))

x <- model.matrix(event ~ . -time,
                  data = train[,-(3:7), , drop = FALSE])

set.seed(1234)
cox_enet_mod_nc <- cv.glmnet(x = x, y = y, family = "cox",
                             # family = "binomial",
                              nfolds = 5,
                 alpha = 0.7)

plot(cox_enet_mod_nc)
```



```
cc_enet_min_nc <- coef(cox_enet_mod_nc, s = cox_enet_mod_nc$lambda.min)

select_vars_enet_nc <- cc_enet_min_nc@Dimnames[[1]][-1][cc_enet_min_nc@i]

selected_coefs_enet_nc <- cc_enet_min_nc@x

names(selected_coefs_enet_nc) <- select_vars_enet_nc

selected_coefs_enet_nc
```

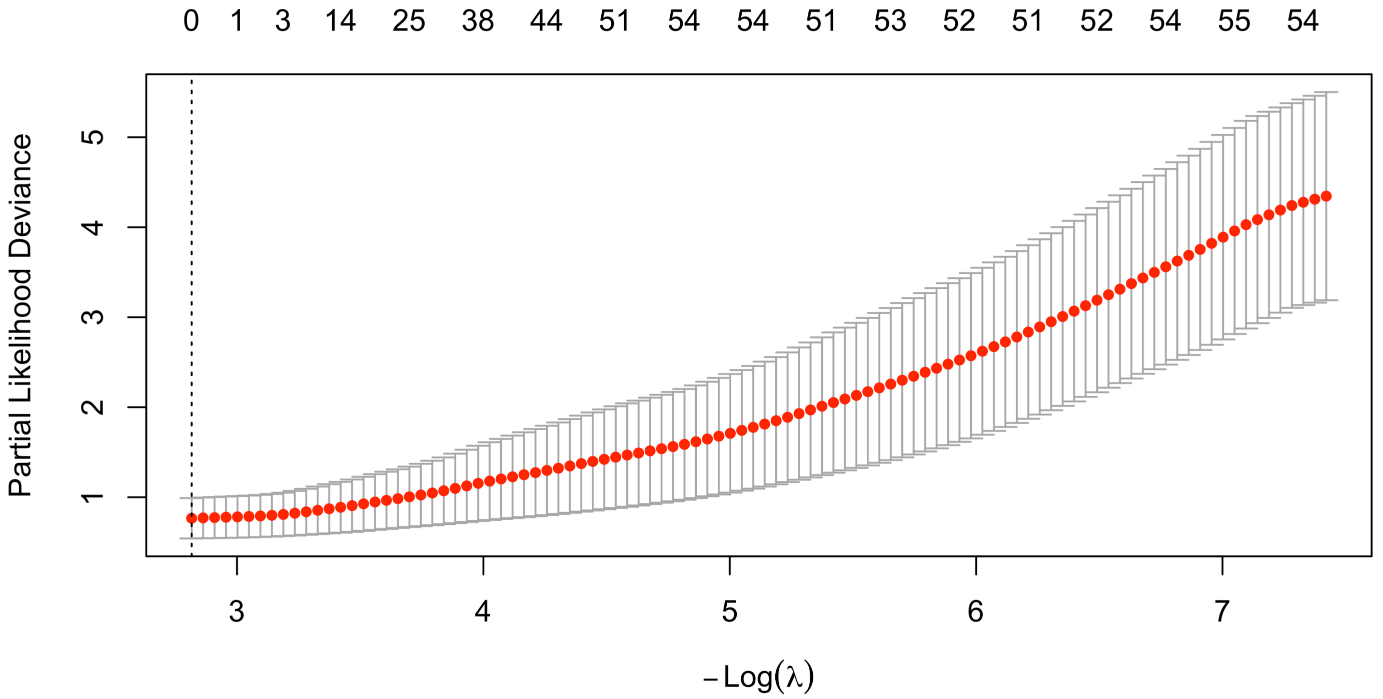
```
#> named numeric(0)
```

```
length(selected_coefs_enet_nc)
```

```
#> [1] 0
```

4.4 Cause-specific Cox models with LASSO

```
y <- Surv(time = train$time,
       event = as.numeric(train$event == 1))
x <- model.matrix(event ~ . -time,
            data = train[,-(3:7), , drop = FALSE])

set.seed(1234)
cox_lasso_mod_nc <- cv.glmnet(x = x, y = y, family = "cox",
              # family = "binomial",
              nfolds = 5,
              thresh = 1e-9,
              maxit = 1e9,
              alpha = 1)

plot(cox_lasso_mod_nc)
```



```
cc_lasso_min_nc <- coef(cox_lasso_mod_nc, s = cox_lasso_mod$lambda.min)

select_vars_lasso_nc <- cc_lasso_min_nc@Dimnames[[1]][-1][cc_lasso_min_nc@i]

selected_coefs_lasso_nc <- cc_lasso_min_nc@x

names(selected_coefs_lasso_nc) <- select_vars_lasso_nc

selected_coefs_lasso_nc
```

```
#>   seq634
#> 0.5562901
```

```
length(selected_coefs_lasso_nc)
```

```
#> [1] 1
```

4.5 Summary

```
# No penalizar variables clínicas.

cbscrip_enet_coefs_nc <- select_coef_cbscrip_enet_nc[!(names(select_coef_cbscrip_enet_nc)
                            c("(Intercept)",
                             "log(time)"))]
```

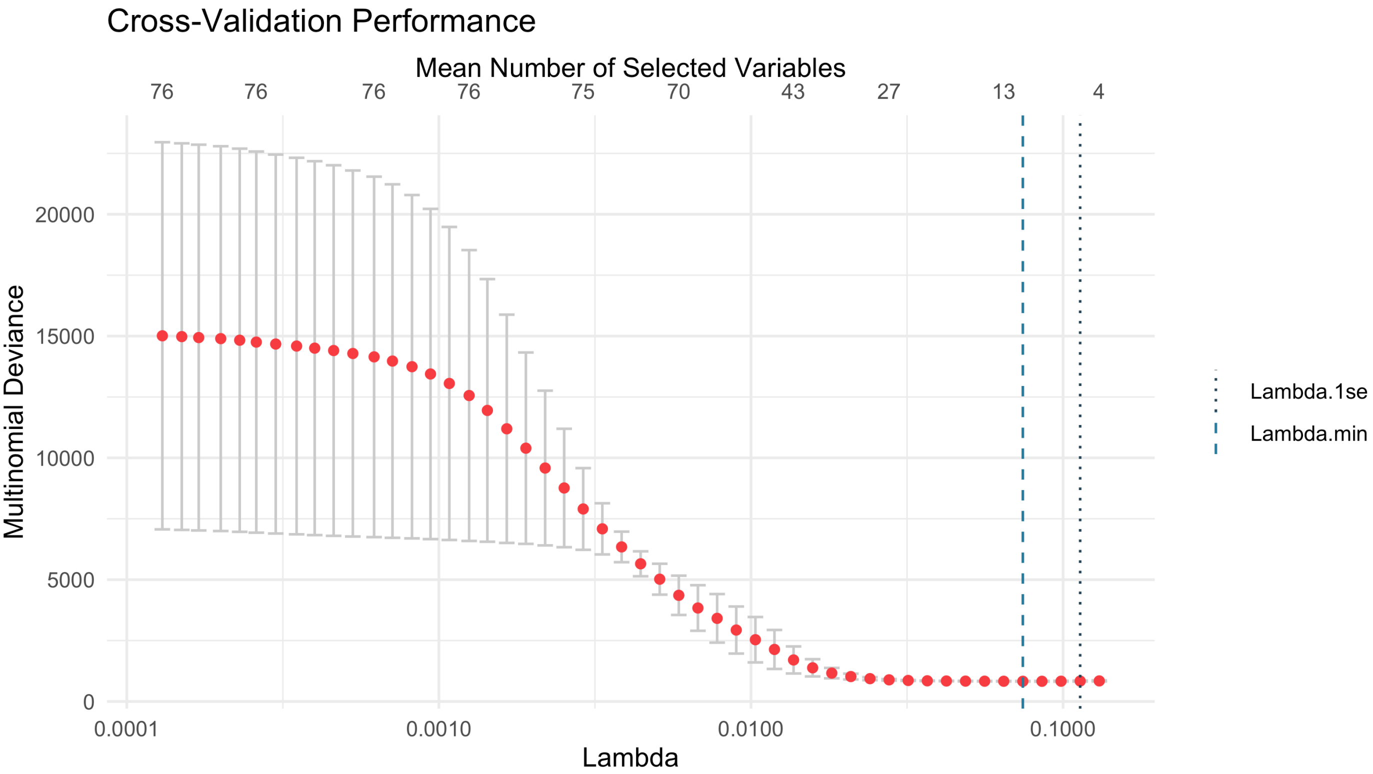
```
cbscrip_scad_coefs_nc <- select_coef_cbscrip_scad_nc[!(names(select_coef_cbscrip_scad_nc)
                            c("(Intercept)",
                             "log(time)"))]

tibble(vars = names(selected_coefs_lasso_nc),
              lasso = selected_coefs_lasso_nc) %>%
  full_join(tibble(vars = names(selected_coefs_enet_nc),
              enet = selected_coefs_enet_nc),
        by = join_by(vars))  %>%
  full_join(tibble(vars = names(cbscrip_enet_coefs_nc),
              cbscrip_enet = cbscrip_enet_coefs_nc),
        by = join_by(vars))  %>%
  full_join(tibble(vars = names(cbscrip_scad_coefs_nc),
              cbscrip_scad = cbscrip_scad_coefs_nc),
        by = join_by(vars)) %>%
    kbl()
```

vars	lasso	enet	cbscrip_enet	cbscrip_scad
seq634	0.5562901	NA	NA	NA

5 Analysis with Subset Proteins

5.1 Multinomial Elastic-net Case-Base Model

```
# Perform cross-validation on the training data
variables_to_select <- c(
 "SEQ1014", "SEQ1038", "SEQ1082", "SEQ1111", "SEQ1164", "SEQ1197",
 "SEQ1225", "SEQ1226", "SEQ1262", "SEQ1330", "SEQ1381", "SEQ1384",
 "SEQ162", "SEQ164", "SEQ183", "SEQ213", "SEQ240", "SEQ251",
 "SEQ265", "SEQ279", "SEQ287", "SEQ34", "SEQ347", "SEQ370",
 "SEQ377", "SEQ410", "SEQ424", "SEQ634", "SEQ681", "SEQ785",
 "SEQ813", "SEQ820", "SEQ833", "SEQ940", "SEQ972", "SEQ973"
)


variables_to_select <- make_clean_names(variables_to_select)


if(save){

  set.seed(1234)
  cv_multinom_enet_sp <- cv_cbSCRIP(
    Surv(time, event) ~ .,
    train[,c("time", "event", variables_to_select), , drop = FALSE],
    alpha = 0.7,
    nfold = 5,
    nlambda = 50,
     fit_fun = mtool_fit_fun,
     ratio = 20)

  plot(cv_multinom_enet_sp)

  saveRDS(cv_multinom_enet_sp, here("paper",
           "results",
           glue("cv_multinom_enet_sp.rds")))

}


cv_multinom_enet_sp <- readRDS(here("paper",
                     "results",
                     glue("cv_multinom_enet_sp.rds")))


plot(cv_multinom_enet_sp)
```



```
cv_multinom_enet_sp$fit.min$coefficients_sparse
```

```
#> 38 x 2 sparse Matrix of class "dgCMatrix"
#>
#> seq1014    .      .
#> seq1038    .      .
#> seq1082    .      .
#> seq1111    .      .
#> seq1164    .      .
#> seq1197    .      .
#> seq1225    .      .
#> seq1226    .      .
#> seq1262    .      .
#> seq1330    .      .
#> seq1381    .      .
#> seq1384    .      .
#> seq162     .      .
#> seq164     .      .
#> seq183     .      .
#> seq213     .      .
#> seq240     .     -0.09401892
#> seq251     .      .
#> seq265     .      0.15566803
#> seq279     .      .
#> seq287     .      .
#> seq34      .      0.14579209
#> seq347     .      .
#> seq370     .      0.02494285
#> seq377     .      0.02395016
#> seq410     .      .
#> seq424     .      .
#> seq634     .      .
#> seq681     .      .
#> seq785     .      .
#> seq813     .      .
#> seq820     .      .
#> seq833     .      0.04373402
#> seq940     .      .
#> seq972     .      0.00359079
#> seq973     .      0.06927822
#> log(time)  -1.538054 -1.46958678
#> (Intercept) -1.406282 -0.66076072
```

```
## Number of non zero parameters (both cases)
sum(!same(cv_multinom_enet_sp$fit.min$coefficients_sparse, 0))
```

```
#> [1] 12
```

```
coefs_1_sp <- cv_multinom_enet_sp$fit.min$coefficients[,1]

# selected variables for event 1.
select_coef_cbscrip_enet_sp <- coefs_1_sp[!same(coefs_1_sp, 0)]
# select_coef_cbscrip_enet <- cv_multinom_enet_sp$fit.min$coefficients[!same(cv_multinom_e


## Number of non zero variables (case 1)
length(select_coef_cbscrip_enet_sp) - 2
```

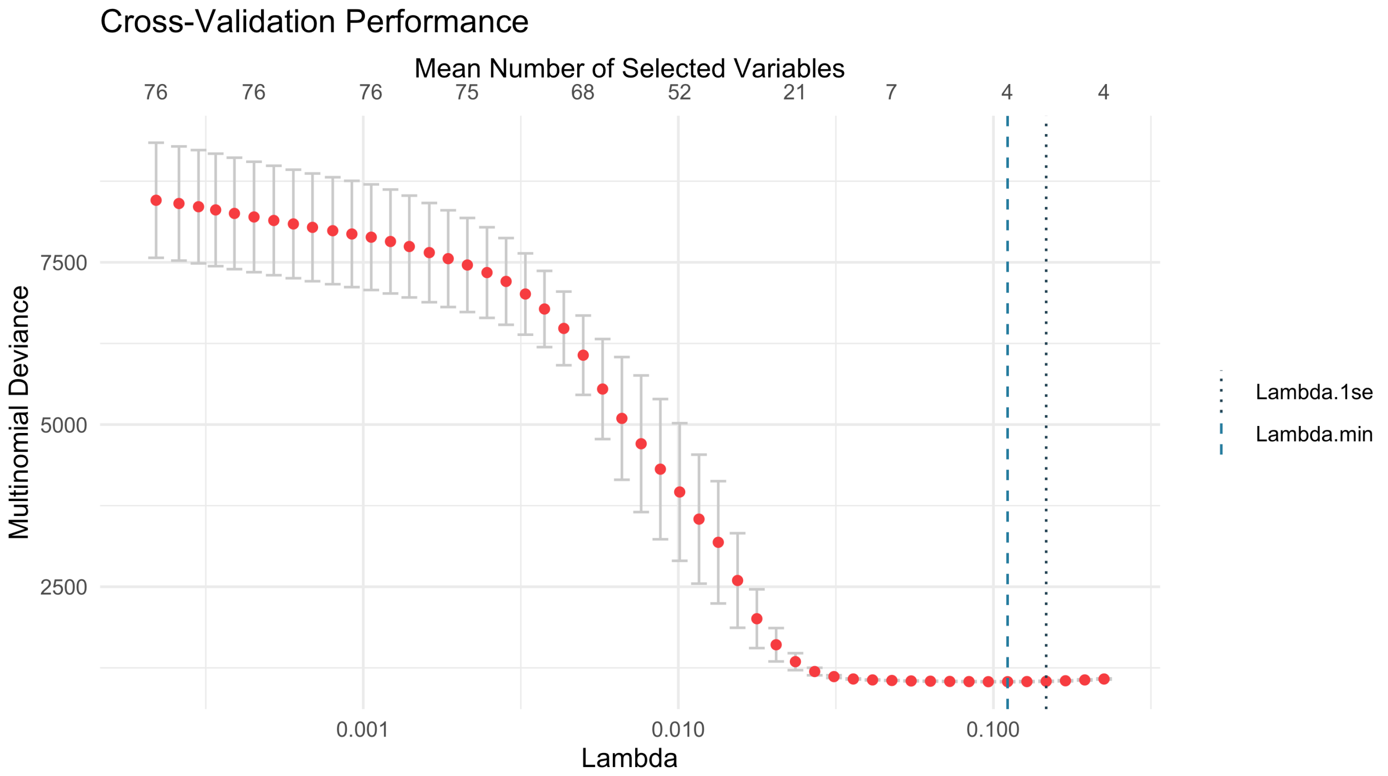
```
#> [1] 0
```

```
select_coef_cbscrip_enet_sp
```

```
#>  log(time) (Intercept)
#>  -1.538054  -1.406282
```

5.2 Multinomial SCAD Case-Base Model

```
# Perform cross-validation on the training data
if(save){

  set.seed(1234)
  cv_multinom_scad_sp <- cv_cbSCRIP(
    Surv(time, event) ~ .,
    train[,c("time", "event", variables_to_select), , drop = FALSE],
    nfold = 5,
    nlambda = 50,
    fit_fun = mtool_fit_fun,
    regularization = "SCAD")

  plot(cv_multinom_scad_sp)

   saveRDS(cv_multinom_scad_sp,
      here("paper",
        "results",
        glue("cv_multinom_scad_sp.rds")))
}

cv_multinom_scad_sp <- readRDS(here("paper",
        "results",
        glue("cv_multinom_scad_sp.rds")))

plot(cv_multinom_scad_sp)
```



```
cv_multinom_scad_sp$fit.min$coefficients
```

```
#>           [,1]     [,2]
#> seq1014   0.000000 0.00000000
#> seq1038   0.000000 0.00000000
#> seq1082   0.000000 0.00000000
#> seq1111   0.000000 0.00000000
#> seq1164   0.000000 0.00000000
#> seq1197   0.000000 0.00000000
#> seq1225   0.000000 0.00000000
#> seq1226   0.000000 0.00000000
#> seq1262   0.000000 0.00000000
#> seq1330   0.000000 0.00000000
#> seq1381   0.000000 0.00000000
#> seq1384   0.000000 0.00000000
#> seq162    0.000000 0.00000000
#> seq164    0.000000 0.00000000
#> seq183    0.000000 0.00000000
#> seq213    0.000000 0.00000000
#> seq240    0.000000 0.00000000
#> seq251    0.000000 0.00000000
#> seq265    0.000000 0.00000000
#> seq279    0.000000 0.00000000
#> seq287    0.000000 0.00000000
#> seq34     0.000000 0.00000000
#> seq347    0.000000 0.00000000
#> seq370    0.000000 0.00000000
#> seq377    0.000000 0.00000000
#> seq410      0.000000  0.00000000
#> seq424      0.000000 -0.02306055
#> seq634      0.000000  0.00000000
#> seq681      0.000000  0.00000000
#> seq785      0.000000  0.00000000
#> seq813      0.000000  0.00000000
#> seq820      0.000000  0.00000000
#> seq833      0.000000  0.00000000
#> seq940      0.000000  0.00000000
#> seq972      0.000000  0.00000000
#> seq973      0.000000  0.00000000
#> log(time)  -1.581013 -1.49299287
#> (Intercept) -1.517253 -1.23603268
```

```
# selected variables for event 1.
coefs_1_scad_sp <- cv_multinom_scad_sp$fit.min$coefficients[,1]

# selected variables for event 1.
select_coef_cbscrip_scad_sp <- coefs_1_scad_sp[!same(coefs_1_scad_sp, 0)]
# select_coef_cbscrip_scad <- cv_multinom_scad_sp$fit.min$coefficients[!same(cv_multinom_s

select_coef_cbscrip_scad_sp
```

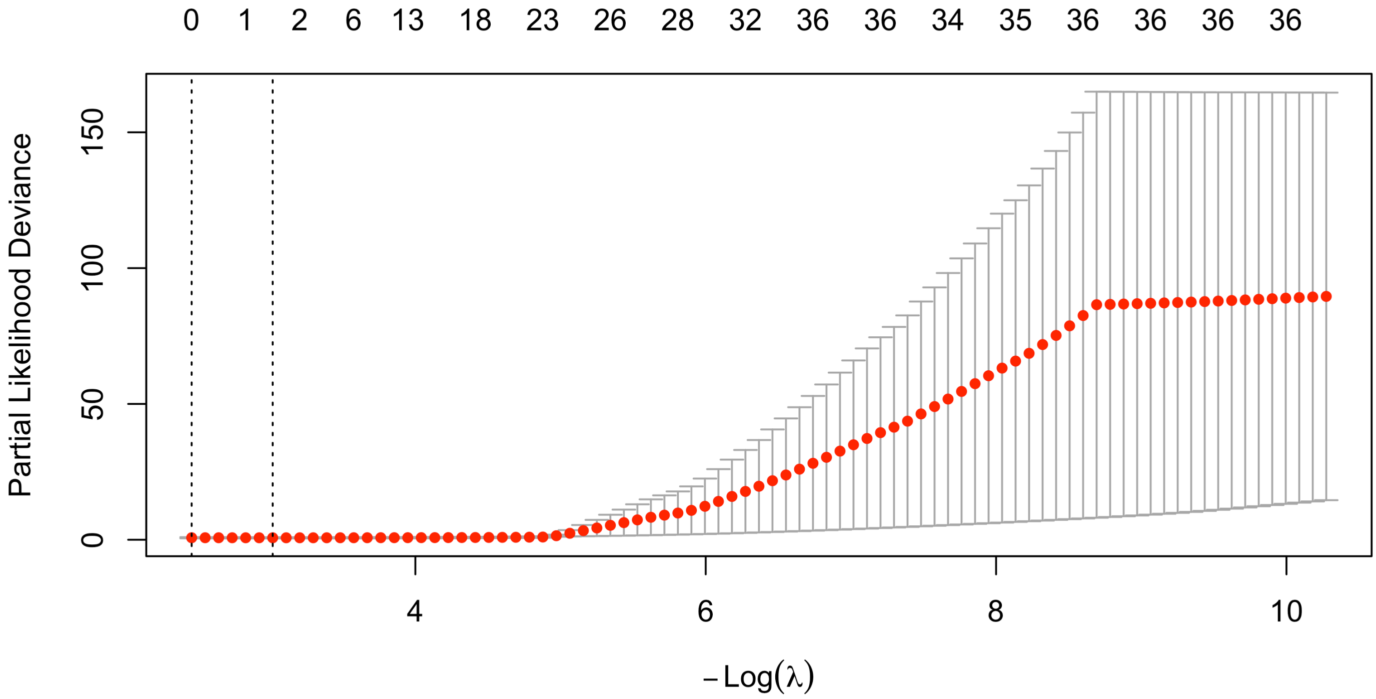
```
#>  log(time) (Intercept)
#>  -1.581013  -1.517253
```

```
## Number of non zero variables
sum(!same(select_coef_cbscrip_scad_sp, 0)) - 2
```

```
#> [1] 0
```

5.3 Cause-specific Cox models with Elastic-net

```
y <- Surv(time = train$time,
      event = as.numeric(train$event == 1))

x <- model.matrix(event ~ . -time,
           data = train[,c("time", "event", variables_to_select), ,
                   drop = FALSE])

set.seed(1234)
cox_enet_mod_sp <- cv.glmnet(x = x, y = y, family = "cox",
                # family = "binomial",
                 nfolds = 5,
                alpha = 0.7)

plot(cox_enet_mod_sp)
```



```
cc_enet_min_sp <- coef(cox_enet_mod_sp, s = cox_enet_mod_sp$lambda.min)

select_vars_enet_sp <- cc_enet_min_sp@Dimnames[[1]][-1][cc_enet_min_sp@i]

selected_coefs_enet_sp <- cc_enet_min_sp@x

names(selected_coefs_enet_sp) <- select_vars_enet_sp

selected_coefs_enet_sp
```

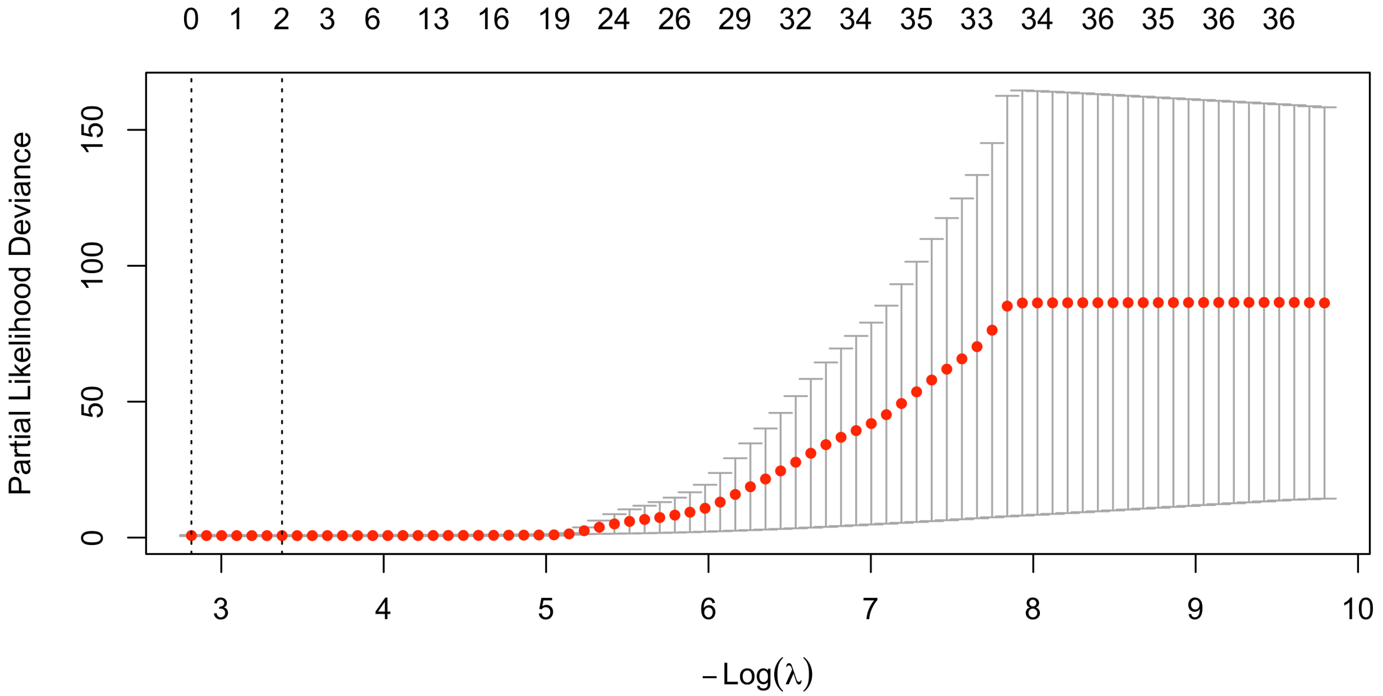
```
#>      seq1330      seq634
#> 0.0008515853 0.9687676008
```

```
length(selected_coefs_enet_sp)
```

```
#> [1] 2
```

5.4 Cause-specific Cox models with LASSO

```
y <- Surv(time = train$time,
          event = as.numeric(train$event == 1))
x <- model.matrix(event ~ . -time,
                  data = train[,c("time", "event", variables_to_select), ,
                               drop = FALSE])

set.seed(1234)
cox_lasso_mod_sp <- cv.glmnet(x = x, y = y, family = "cox",
                    # family = "binomial",
                    nfolds = 5,
                    thresh = 1e-9,
                    maxit = 1e9,
             alpha = 1)

plot(cox_lasso_mod_sp)
```



```
cc_lasso_min_sp <- coef(cox_lasso_mod_sp, s = cox_lasso_mod_sp$lambda.min)

select_vars_lasso_sp <- cc_lasso_min_sp@Dimnames[[1]][-1][cc_lasso_min_sp@i]

selected_coefs_lasso_sp <- cc_lasso_min_sp@x

names(selected_coefs_lasso_sp) <- select_vars_lasso_sp

selected_coefs_lasso_sp
```

```
#>    seq1330    seq634
#> 0.01847636 1.18500434
```

```
length(selected_coefs_lasso_sp)
```

```
#> [1] 2
```

5.5 Summary

```
# No penalizar variables clínicas.

cbscrip_enet_coefs_sp <- select_coef_cbscrip_enet_sp[!(names(select_coef_cbscrip_enet_sp)
                            c("(Intercept)",
                             "log(time)"))]
cbscrip_scad_coefs_sp <- select_coef_cbscrip_scad_sp[!(names(select_coef_cbscrip_scad_sp)
                            c("(Intercept)",
                              "log(time)"))]

tibble(vars = names(selected_coefs_lasso_sp),
              lasso = selected_coefs_lasso_sp) %>%
  full_join(tibble(vars = names(selected_coefs_enet_sp),
              enet = selected_coefs_enet_sp),
         by = join_by(vars))  %>%
  full_join(tibble(vars = names(cbscrip_enet_coefs_sp),
              cbscrip_enet = cbscrip_enet_coefs_sp),
         by = join_by(vars))  %>%
  full_join(tibble(vars = names(cbscrip_scad_coefs_sp),
              cbscrip_scad = cbscrip_scad_coefs_sp),
         by = join_by(vars)) %>%
    kbl()
```

vars	lasso	enet	cbscrip_enet	cbscrip_scad
seq1330	0.0184764	0.0008516	NA	NA
seq634	1.1850043	0.9687676	NA	NA

6 Analysis Top Variables

6.1 Multinomial Elastic-net Case-Base Model

```
# Perform cross-validation on the training data

if(save){

  set.seed(1234)
  cv_multinom_enet_top <- cbSCRIP(
    Surv(time, event) ~ .,
    train[,-(3:7), , drop = FALSE],
    alpha = 0.7,
    nlambda = 50,
    fit_fun = mtool_fit_fun)

  saveRDS(cv_multinom_enet_top, here("paper",
          "results",
          glue("cv_multinom_enet_top.rds")))

}


cv_multinom_enet_top <- readRDS(here("paper",
                     "results",
                     glue("cv_multinom_enet_top.rds")))
```

```
non_cero_top_enet <- map(cv_multinom_enet_top$models_info,
                ~ sum(!same(.x$coefficients[,1], 0)))

top_50_cbscrip_enet <-  which.min(abs(map_int(non_cero_top_enet, ~.x)-30))

coeffs_top_50_cbscrip_enet <- cv_multinom_enet_top$models_info[[top_50_cbscrip_enet]]$coe

# selected variables for event 1.
select_coef_cbscrip_enet_top <- coeffs_top_50_cbscrip_enet[!same(coeffs_top_50_cbscrip_en

## Number of non zero variables
length(select_coef_cbscrip_enet_top) - 2
```

```
#> [1] 28
```

```
select_coef_cbscrip_enet_top
```

```
#>    seq1017    seq1092     seq111    seq1178     seq121    seq1277
#>  0.01367908 -0.10228849 -0.15471508 -0.07555757 -0.10965223 -0.08799742
#>    seq1317   seq1384_2      seq17     seq386     seq394     seq414
#> -0.01934672 -0.03001052  0.10507870  0.00338641 -0.03643190 -0.28494498
#>     seq500     seq508     seq560     seq580     seq588     seq592
#>  0.02186111 -0.02619492 -0.16229464  0.21585887  0.04348343  0.07259799
#>     seq627     seq634     seq758     seq846     seq850     seq862
#> -0.08172483  0.72750698 -0.01382223  0.01053330  0.07210274 -0.15633116
#>     seq913     seq934     seq961     seq992   log(time) (Intercept)
#> -0.06014896 -0.13788020 -0.02435717 -0.12216939  0.33263745 -8.07542891
```

6.2 Multinomial SCAD Case-Base Model

```
# Perform cross-validation on the training data
if(save){

  set.seed(1234)
  cv_multinom_scad_top <- cbSCRIP(
    Surv(time, event) ~ .,
    train[,-(3:7), , drop = FALSE],
    nlambda = 50,
    fit_fun = mtool_fit_fun,
    regularization = "SCAD"
  )


    saveRDS(cv_multinom_scad_top,
       here("paper",
          "results",
          glue("cv_multinom_scad_top.rds")))
}
```

```
cv_multinom_scad_top <- readRDS(here("paper",
          "results",
          glue("cv_multinom_scad_top.rds")))

non_cero_top_scad <- map(cv_multinom_scad_top$models_info,
                ~ sum(!same(.x$coefficients[,1], 0)))

top_50_cbscrip_scad <-  which.min(abs(map_int(non_cero_top_scad, ~.x)-30))

coeffs_top_50_cbscrip_scad <- cv_multinom_scad_top$models_info[[top_50_cbscrip_scad]]$coe

# selected variables for event 1.
select_coef_cbscrip_scad_top <- coeffs_top_50_cbscrip_scad[!same(coeffs_top_50_cbscrip_sca

## Number of non zero variables
length(select_coef_cbscrip_scad_top) - 2
```

```
#> [1] 30
```

```
select_coef_cbscrip_scad_top
```

```
#>    seq1020    seq1115     seq121    seq1213    seq1277    seq1307
#> -0.17754058 -0.22223748 -0.20144088  0.31087523 -0.34415957  0.06318887
#>    seq1317    seq1330    seq1364    seq1383   seq1384_2     seq147
#> -0.25908471  0.12662384 -0.18953123  0.24590489 -0.60234012 -0.38837647
#>     seq173     seq283     seq360     seq414     seq419     seq508
#>  0.31410479 -0.01007387  0.39437767 -0.43990814  0.11729992 -0.46238739
#>     seq560     seq599     seq634     seq749     seq758     seq842
#> -0.20866945  0.27410384  2.04373661 -0.17276542 -0.18496491 -0.21787961
#>     seq862     seq879     seq909     seq913     seq948     seq960
#> -0.32817188 -0.13941492 -0.22377179 -0.21422067 -0.19769871 -0.09014079
#>   log(time) (Intercept)
#>  0.28549695 -8.94623620
```

6.3 Cause-specific Cox models with Elastic-net

```
y <- Surv(time = train$time,
      event = as.numeric(train$event == 1))

x <- model.matrix(event ~ . -time,
           data = train[,-(3:7), , drop = FALSE])

set.seed(1234)
cox_enet_mod_top <- glmnet(x = x, y = y, family = "cox",
                 # family = "binomial",
                  nfolds = 5,
                 alpha = 0.7)

grid <- cox_enet_mod_top$lambda
```

```r
non_zero_enet <- map_dbl(grid,
             ~sum(!same(coef(cox_enet_mod_top, s = .x), 0)))

cc_enet_min_top <- coef(cox_enet_mod_top, s = grid[which.min(abs(non_zero_enet-30))])

select_vars_enet_top <- cc_enet_min_top@Dimnames[[1]][-1][cc_enet_min_top@i]

selected_coefs_enet_top <- cc_enet_min_top@x

names(selected_coefs_enet_top) <- select_vars_enet_top

selected_coefs_enet_top
```

```
#>       seq1092        seq111       seq1178        seq121       seq1317
#> -0.0499624653 -0.4175532211 -0.3482580876 -0.2093640444 -0.0080463157
#>       seq1330     seq1384_2        seq147         seq17        seq257
#>  0.0657892683 -0.0111091017 -0.1380388499  0.3141901919 -0.0095793214
#>        seq394        seq500        seq508        seq560        seq580
#> -0.0558383965  0.1101898041 -0.1074258126 -0.2419927839  0.3243705417
#>        seq588        seq592        seq627        seq634        seq644
#>  0.0794076927  0.0003725202 -0.1273008028  1.1426884796  0.1097774482
#>        seq758        seq784        seq850        seq862        seq894
#> -0.0507257587 -0.1083157001  0.1288298290 -0.3139509048 -0.0513106444
#>        seq909        seq934        seq948        seq961        seq992
#> -0.2967764356 -0.1085200394 -0.0096372871 -0.0617258605 -0.3266030308
```

```r
length(selected_coefs_enet_top)
```

```
#> [1] 30
```

6.4 Cause-specific Cox models with LASSO

```r
y <- Surv(time = train$time,
      event = as.numeric(train$event == 1))

x <- model.matrix(event ~ . -time,
            data = train[,-(3:7), , drop = FALSE])

set.seed(1234)
cox_lasso_mod_top <- glmnet(x = x, y = y, family = "cox",
                 # family = "binomial",
                  nfolds = 5,
                 alpha = 1)

grid <- cox_lasso_mod_top$lambda
```

```
non_zero_lasso <- map_dbl(grid,
            ~sum(!same(coef(cox_lasso_mod_top, s = .x), 0)))

cc_lasso_min_top <- coef(cox_lasso_mod_top, s = grid[which.min(abs(non_zero_lasso-30))])

select_vars_lasso_top <- cc_lasso_min_top@Dimnames[[1]][-1][cc_lasso_min_top@i]

selected_coefs_lasso_top <- cc_lasso_min_top@x

names(selected_coefs_lasso_top) <- select_vars_lasso_top

selected_coefs_lasso_top
```

```
#>     seq111    seq1178     seq121    seq1308    seq1311    seq1317
#> -0.78504392 -0.52636768 -0.23504764 -0.01295189  0.02710873 -0.03437834
#>    seq1330     seq147      seq17     seq248     seq394     seq414
#>  0.14541072 -0.19337211  0.44156579 -0.04456659 -0.11121302 -0.02493558
#>     seq500     seq508     seq560     seq580     seq588     seq627
#>  0.22988032 -0.44619675 -0.37233182  0.40231567  0.07948048 -0.15526703
#>     seq634     seq644     seq758     seq784     seq850     seq862
#>  1.66920482  0.16523140 -0.14446074 -0.20446203  0.10266160 -0.40903617
#>     seq894     seq909     seq934     seq948     seq961     seq992
#> -0.09740454 -0.19587478 -0.27865313 -0.02360715 -0.12715098 -0.40846532
```

```
length(selected_coefs_lasso_top)
```

```
#> [1] 30
```

6.5 Summary

```
# No penalizar variables clínicas.

cbscrip_enet_coefs_top <- select_coef_cbscrip_enet_top[!(names(select_coef_cbscrip_enet_t
                           c("(Intercept)",
                            "log(time)"))]
cbscrip_scad_coefs_top <- select_coef_cbscrip_scad_top[!(names(select_coef_cbscrip_scad_t
                           c("(Intercept)",
                            "log(time)"))]

tibble(vars = names(cbscrip_enet_coefs_top),
    cbscrip_enet = cbscrip_enet_coefs_top) %>%
  full_join(tibble(vars = names(cbscrip_scad_coefs_top),
            cbscrip_scad = cbscrip_scad_coefs_top),
        by = join_by(vars)) %>%
  full_join(tibble(vars = names(selected_coefs_lasso_top),
            lasso = selected_coefs_lasso_top),
        by = join_by(vars))  %>%
  full_join(tibble(vars = names(selected_coefs_enet_top),
            enet = selected_coefs_enet_top),
       by = join_by(vars)) %>%
  kbl()
```

vars	cbscrip_enet	cbscrip_scad	lasso	enet
seq1017	0.0136791	NA	NA	NA
seq1092	-0.1022885	NA	NA	-0.0499625
seq111	-0.1547151	NA	-0.7850439	-0.4175532
seq1178	-0.0755576	NA	-0.5263677	-0.3482581
seq121	-0.1096522	-0.2014409	-0.2350476	-0.2093640
seq1277	-0.0879974	-0.3441596	NA	NA
seq1317	-0.0193467	-0.2590847	-0.0343783	-0.0080463
seq1384_2	-0.0300105	-0.6023401	NA	-0.0111091
seq17	0.1050787	NA	0.4415658	0.3141902
seq386	0.0033864	NA	NA	NA
seq394	-0.0364319	NA	-0.1112130	-0.0558384
seq414	-0.2849450	-0.4399081	-0.0249356	NA
seq500	0.0218611	NA	0.2298803	0.1101898
seq508	-0.0261949	-0.4623874	-0.4461967	-0.1074258
seq560	-0.1622946	-0.2086694	-0.3723318	-0.2419928
seq580	0.2158589	NA	0.4023157	0.3243705
seq588	0.0434834	NA	0.0794805	0.0794077
seq592	0.0725980	NA	NA	0.0003725
seq627	-0.0817248	NA	-0.1552670	-0.1273008
seq634	0.7275070	2.0437366	1.6692048	1.1426885
seq758	-0.0138222	-0.1849649	-0.1444607	-0.0507258
seq846	0.0105333	NA	NA	NA
seq850	0.0721027	NA	0.1026616	0.1288298
seq862	-0.1563312	-0.3281719	-0.4090362	-0.3139509
seq913	-0.0601490	-0.2142207	NA	NA
seq934	-0.1378802	NA	-0.2786531	-0.1085200
seq961	-0.0243572	NA	-0.1271510	-0.0617259
seq992	-0.1221694	NA	-0.4084653	-0.3266030
seq1020	NA	-0.1775406	NA	NA
seq1115	NA	-0.2222375	NA	NA
seq1213	NA	0.3108752	NA	NA
seq1307	NA	0.0631889	NA	NA
seq1330	NA	0.1266238	0.1454107	0.0657893
seq1364	NA	-0.1895312	NA	NA
seq1383	NA	0.2459049	NA	NA
seq147	NA	-0.3883765	-0.1933721	-0.1380388
seq173	NA	0.3141048	NA	NA
seq283	NA	-0.0100739	NA	NA
seq360	NA	0.3943777	NA	NA
seq419	NA	0.1172999	NA	NA
seq599	NA	0.2741038	NA	NA
seq749	NA	-0.1727654	NA	NA
seq842	NA	-0.2178796	NA	NA
seq879	NA	-0.1394149	NA	NA
seq909	NA	-0.2237718	-0.1958748	-0.2967764
seq948	NA	-0.1976987	-0.0236071	-0.0096373
seq960	NA	-0.0901408	NA	NA
seq1308	NA	NA	-0.0129519	NA

References

Dyrskjøt, Lars, Mogens Kruhøffer, Thomas Thykjaer, Niels Marcussen, Jens L. Jensen, Klaus Møller, and Torben F. Ørntoft. 2004. "Gene Expression in the Urinary Bladder: A Common Carcinoma in Situ Gene Expression Signature Exists Disregarding Histopathological Classification." *Cancer Research* 64 (11): 4040–48. https://doi.org/10.1158/0008-5472.CAN-03-3620.

Ke, Chenlu, Dipankar Bandyopadhyay, and Devanand Sarkar. 2023. "Gene Screening for Prognosis of Non-Muscle-Invasive Bladder Carcinoma Under Competing Risks Endpoints." *Cancers* 15 (2): 379. https://doi.org/10.3390/cancers15020379.

Tapak, Leili, Massoud Saidijam, Majid Sadeghifar, Jalal Poorolajal, and Hossein Mahjub. 2015. "Competing Risks Data Analysis with High-Dimensional Covariates: An Application in Bladder Cancer." *Genomics, Proteomics & Bioinformatics* 13 (3): 169–76. https://doi.org/10.1016/j.gpb.2015.04.001.